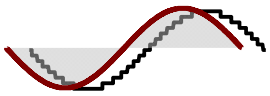
EE315B
VLSI Data Conversion Circuits

- Autumn 2011 -

Katelijn Vleugels
Stanford University

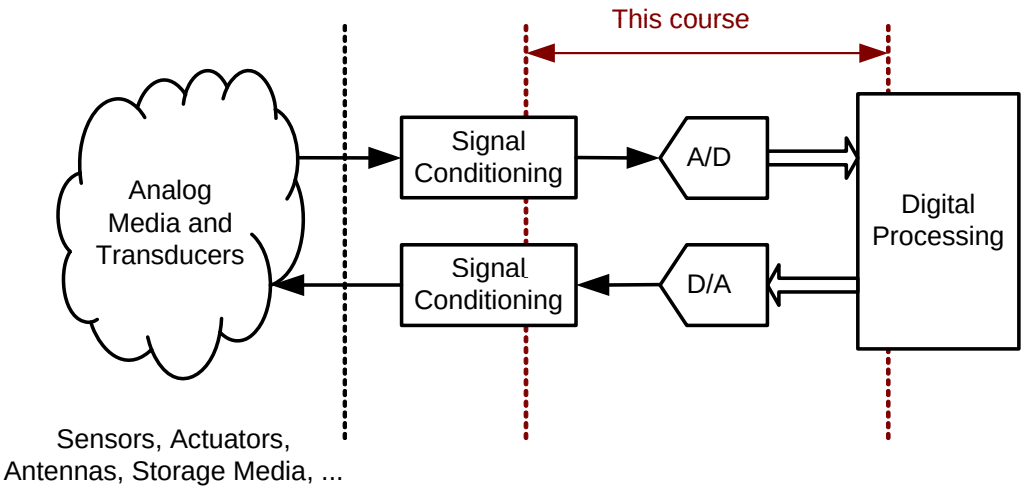
Table of Contents

Introduction



Katelijn Vleugels
Stanford University

Motivation (1)



Motivation (2)

- Benefits of digital signal processing
  - Reduced sensitivity to "analog" noise
  - Enhanced functionality and flexibility
  - Amenable to automated design & test
  - Direct benefit from the scaling of VLSI technology
  - "Arbitrary" precision
- Issues
  - Data converters are difficult to design
    - Especially due to ever-increasing performance requirements
  - Data converters often present a performance bottleneck
    - Speed, resolution or power dissipation of the A/D or D/A converter can limit overall system performance

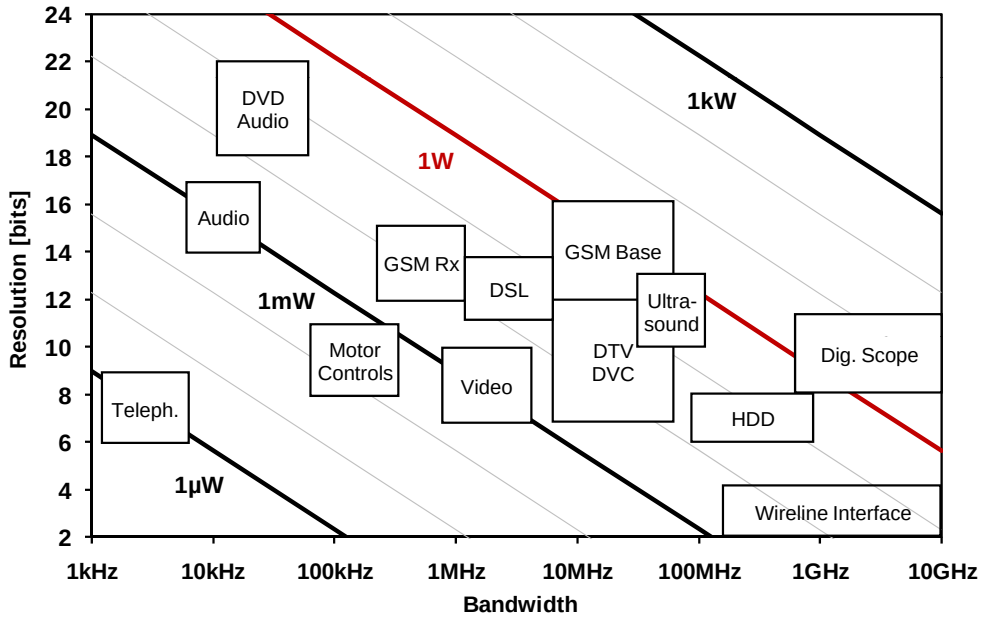
Data Converter Applications (1)

- Consumer electronics
  - Audio, TV, Video
  - Digital Cameras
  - Automotive control
  - Appliances
  - Toys
- Communications
  - Mobile Phones
  - Personal Data Assistants
  - Wireless Transceivers
  - Routers, Modems

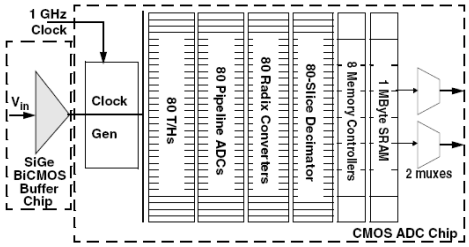
Data Converter Applications (2)

- Computing and Control
  - Storage media
  - Sound Cards
  - Data acquisition cards
- Instrumentation
  - Lab bench equipment
  - Semiconductor test equipment
  - Scientific equipment
  - Medical equipment

A/D Converter Application Space

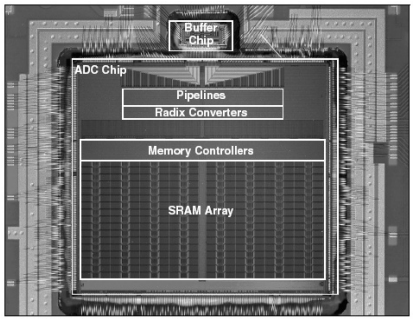


Example 1



[Poulton, ISSCC 2003]

- High performance digital oscilloscopes rely on extremely high performance ADCs
- Example
  - 20 GSample/s, 8-bit ADC
  - 10 W Power dissipation



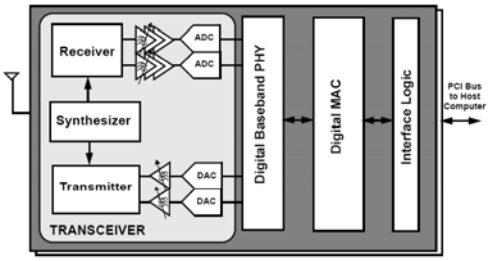
Example 2

- A typical cell phone contains:
  - 4 Rx ADCs
  - 4 Tx DACs

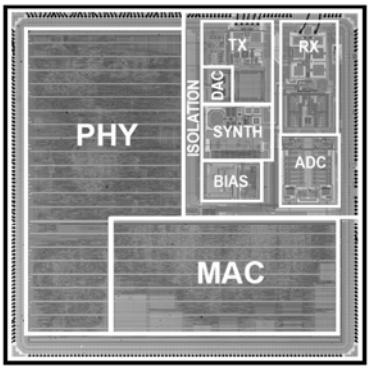
  } Dual Standard, I/Q
  - 3 Auxiliary ADCs
  - 8 Auxiliary DACs

  } Audio, Tx/Rx power control, Battery charge control, display, ...
- A total of 19 data converters!

Example 3

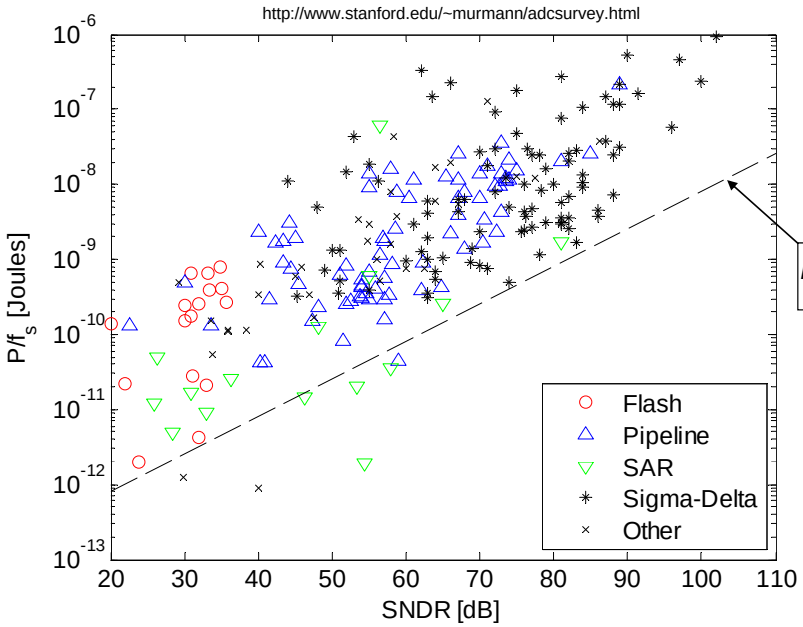


[Mehta, ISSCC2005]



- Low-cost, single chip solutions require embedded data conversion
- Example: 802.11g Wireless LAN chip
  - 2x 11-bit DAC, 176 MSamples/s
  - 2x 9-bit ADC, 80 MSamples/s

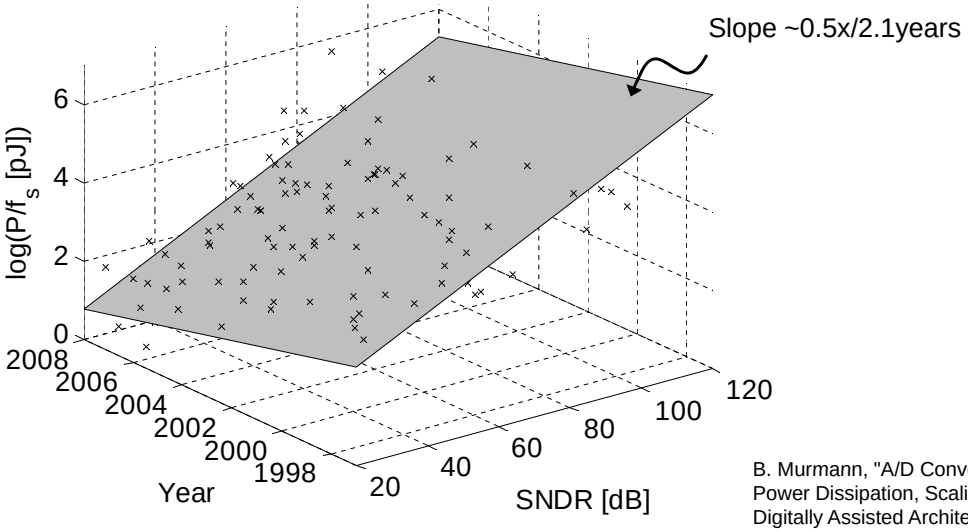
Survey Data (ISSCC & VLSI 1997-2008)



$$FOM = \frac{P}{f_s \cdot 2^{ENOB}} = 100\,\frac{\text{fJ}}{\text{conv-step}}$$

$$ENOB = \frac{SNDR[dB] - 1.76}{6.02}$$

ADC Energy Trend



B. Murmann, "A/D Converter Trends: Power Dissipation, Scaling and Digitally Assisted Architectures," *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, pp. 105-112, Sept. 2008.

- Reduction of ~30x over past 10 years (!)

Course Objective

- Acquire a thorough understanding of the basic principles and challenges in data converter design
  - Focus on concepts that are unlikely to expire within the next decade
  - Preparation for further study of state-of-the-art "fine-tuned" realizations
- Strategy
  - Acquire breadth via a complete system walkthrough and a survey of existing architectures
  - Acquire depth through a midterm project that entails design and thorough characterization of a specific circuit example in modern technology

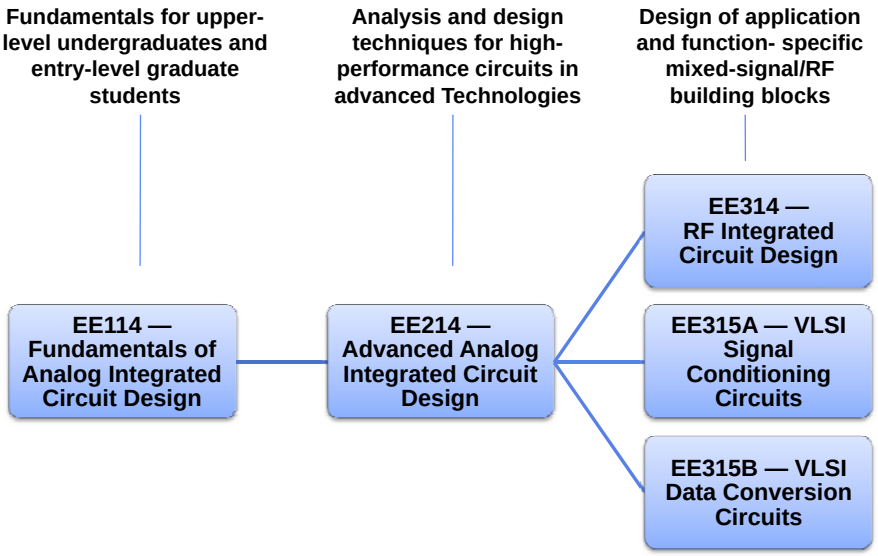
Staff and Website

- Teaching assistants
  - See web
- Administrative support
  - Ann Guerra, CIS 207
- Lecture videos are provided on the web, but please come to class to keep the discussion intercative
- Web page: http://ccnet.stanford.edu/ee315b
  - Check regularly, especially the "bulletin board" section
  - Only enrolled students can register for ccnet access
    - We synchronize the ccnet database with axess.stanford.edu manually, ~ once per day during first week of instruction

Preparation

- Course prerequisites
  - EE214 or equivalent
    - Device physics and models
    - Transistor level analog circuits, elementary gain stages
    - Frequency response, feedback, noise
  - Prior exposure to Spice, Matlab
  - Basic signals and systems
  - Probablity
- Please talk to me if you are not sure if you have the required background

Analog Circuit Sequence



Assignments

- Homework: (20%)
  - Handed out on Tue, due following Tue after lecture (1 pm)
  - Lowest HW score is dropped in final grade calculation
- Midterm Project: (40%)
  - Transistor level design and simulation of a data converter sub-block (no layout)
  - Prepare a project report in the format and style of an IEEE journal paper
- Final Exam (40%)

Honor Code

- Please remember you are bound by the honor code
  - I will trust you not to cheat
  - I will try not to tempt you
- But if you are found cheating it is very serious
  - There is a formal hearing
  - You can be thrown out of Stanford
- Save yourself and me a huge hassle and be honest
- For more info
  - http://www.stanford.edu/dept/vpsa/judicialaffairs/guiding/pdf/honorcode.pdf

Tools and Technology

- Primary tools
  - Cadence Virtuoso Schematic Editor
  - Cadence Virtuoso Analog Design Environment
  - Cadence SpectreRF simulator
  - You can use your own tools/setups “at own risk“
- Getting started
  - Read tutorials and setup info provided in the CAD section of the course website
- EE315A/B Technology
  - 0.18-μm CMOS
  - BSIM3v3 models provided under /usr/class/ee315b/models

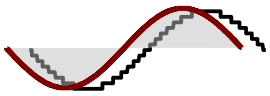
Reference Books

- Gustavsson, Wikner, Tan, *CMOS Data Converters for Communications,* Kluwer, 2000.
- A. Rodríguez-Vázquez, F. Medeiro, and E. Janssens, *CMOS Telecom Data Converters*, Kluwer Academic Publishers, 2003.
- B. Razavi, *Data Conversion System Design*, IEEE Press, 1995.
- R. Schreier, G. Temes, *Understanding Delta-Sigma Data Converters*, Wiley-IEEE Press, 2004.
- R. v. d. Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, 2nd ed., Kluwer, 2003.
- J. G. Proakis, D. G. Manolakis, *Digital Signal Processing*, Prentice Hall, 1995.

Course Topics

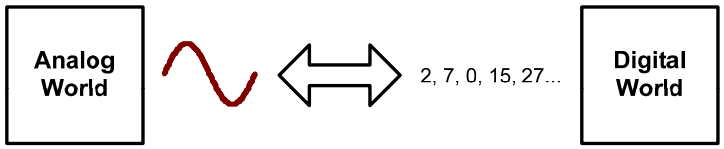
- Ideal sampling, reconstruction and quantization
- Sampling circuits
- Voltage comparators
- Nyquist-rate ADCs and DACs
- Oversampled ADCs and DACs
- Data converter performance trends and limits
- Data converter testing

Sampling, Quantization, Reconstruction



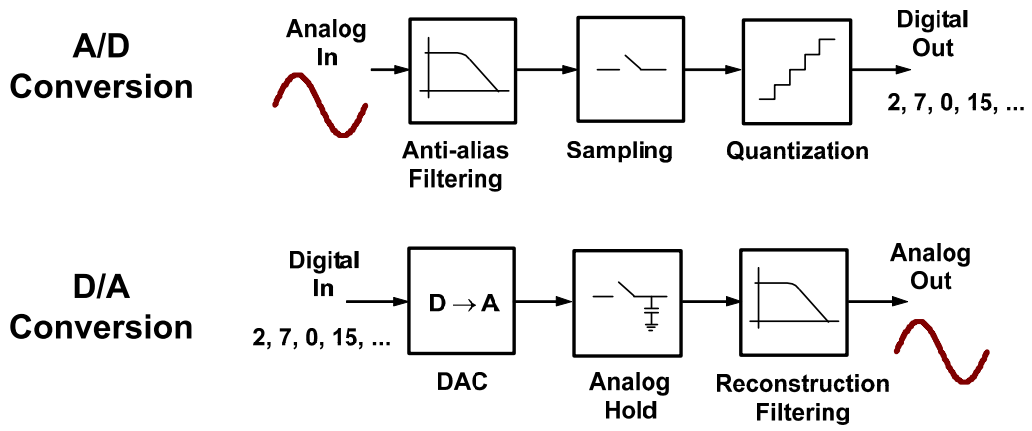
Katelijn Vleugels
Stanford University

The Data Conversion Problem



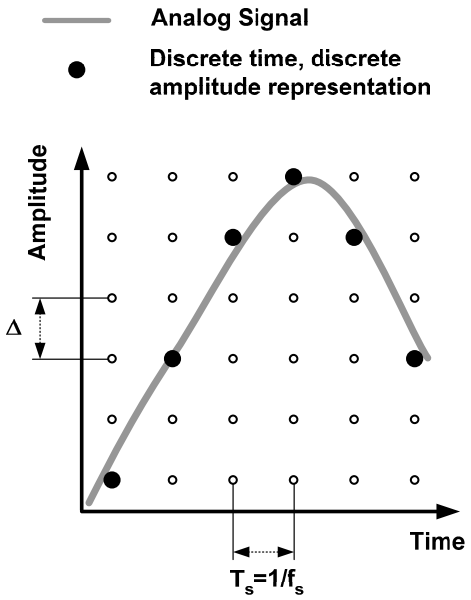
- Real world signals
  - Continuous time, continuous amplitude
- Digital abstraction
  - Discrete time, discrete amplitude
- Two problems
  - How to discretize in time and amplitude
    - A/D conversion
  - How to "undescretize" in time and amplitude
    - D/A conversion

Overview



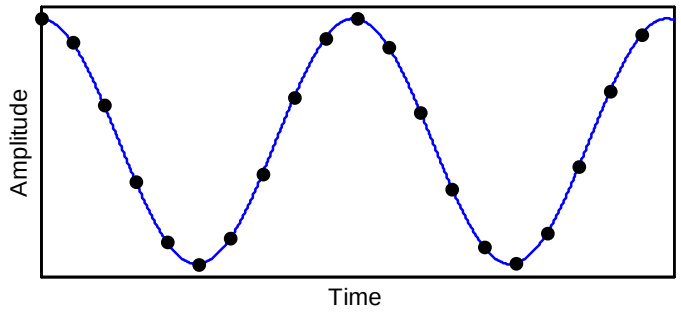
- We'll fist look at these building blocks from a functional, "black box" perspective
  - Refine later and look at implementations

Uniform Sampling and Quantization



- Most common way of performing A/D conversion
  - Sample signal uniformly in time
  - Quantize signal uniformly in amplitude
- Key questions
  - How fast do we need to sample?
    - Must avoid "aliasing"
  - How much "noise" is added due to amplitude quantization?

Aliasing Example (1)



$$f_s = \frac{1}{T_s} = 1000kHz$$

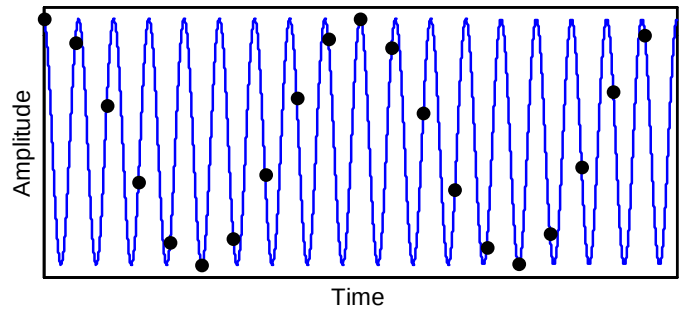
$$f_{sig} = 101kHz$$

$$v_{sig}(t) = \cos(2\pi \cdot f_{in} \cdot t) \Rightarrow v_{sig}(n) = \cos\left(2\pi \cdot \frac{f_{in}}{f_s} \cdot n\right)$$

$$t \rightarrow n \cdot T_s = \frac{n}{f_s}$$

$$= \cos\left(2\pi \cdot \frac{101}{1000} \cdot n\right)$$

Aliasing Example (2)

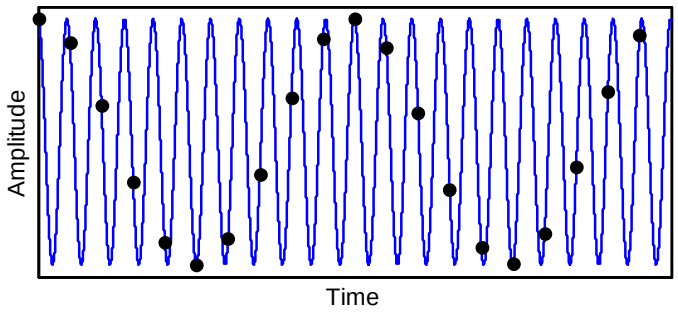


$$f_s = \frac{1}{T_s} = 1000kHz$$

$$f_{sig} = 899kHz$$

$$v_{sig}(n) = \cos\left(2\pi \cdot \frac{899}{1000} \cdot n\right) = \cos\left(2\pi \cdot \left[\frac{899}{1000} - 1\right] \cdot n\right) = \cos\left(2\pi \cdot \frac{101}{1000} \cdot n\right)$$

Aliasing Example (3)

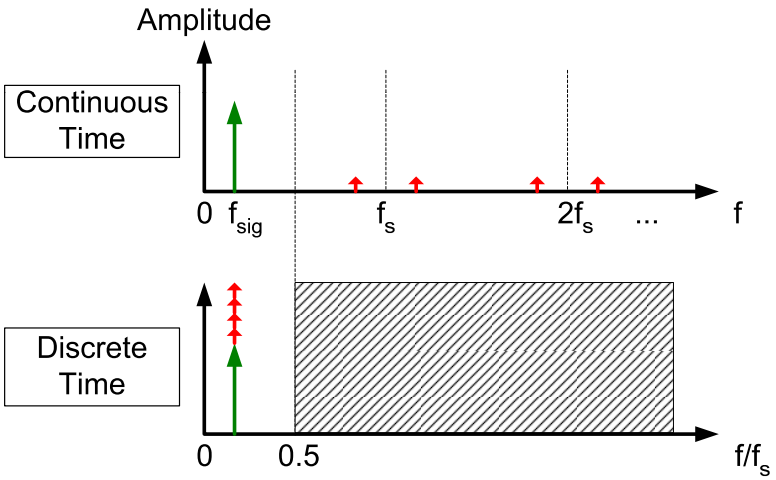


$$f_s = \frac{1}{T_s} = 1000kHz$$

$$f_{sig} = 1101kHz$$

$$v_{sig}(n) = cos\left(2\pi \cdot \frac{1101}{1000} \cdot n\right) = cos\left(2\pi \cdot \left[\frac{1101}{1000} - 1\right] \cdot n\right) = cos\left(2\pi \cdot \frac{101}{1000} \cdot n\right)$$

Consequence



- The frequencies f_{sig} and $N \cdot f_s \pm f_{sig}$ (N integer), are indistinguishable in the discrete time domain

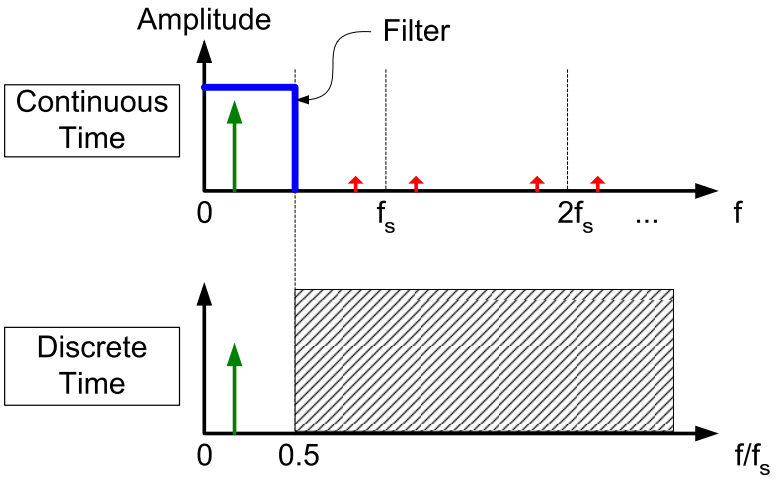
Sampling Theorem

- In order to prevent aliasing, we need

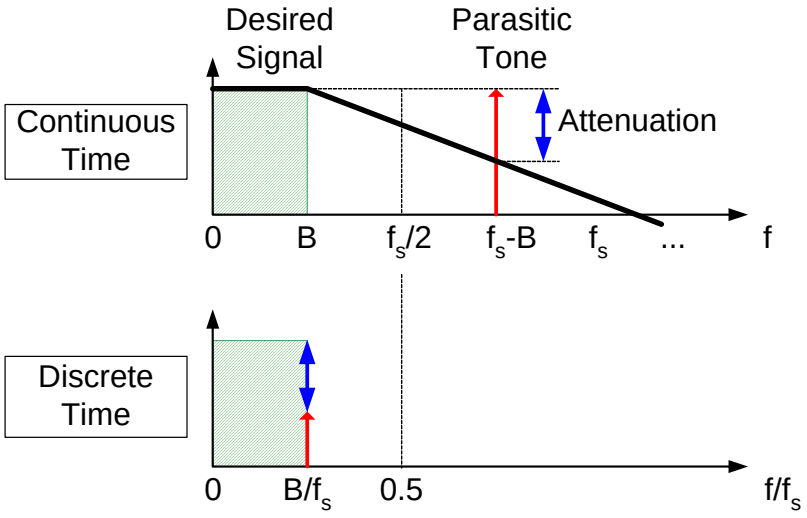
$$f_{sig,max} < \frac{f_s}{2}$$

- The sampling rate $f_s=2 \cdot f_{sig,max}$ is called the Nyquist rate
- Two possibilities
  - Sample fast enough to cover all spectral components, including "parasitic" ones outside band of interest
  - Limit $f_{sig,max}$ through filtering

Brick Wall Anti-Alias Filter

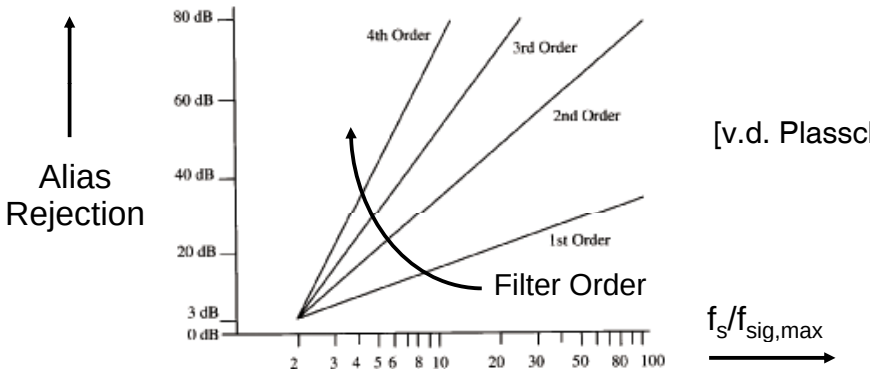


Practical Anti-Alias Filter



- Need to sample faster than Nyquist rate to get good attenuation
  - "Oversampling"

How much Oversampling?



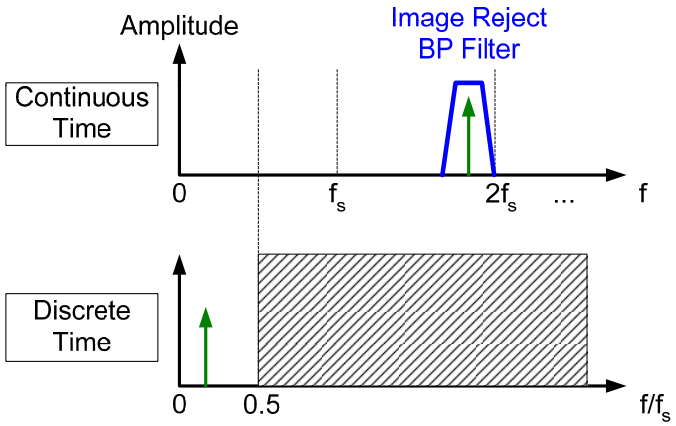
[v.d. Plassche, p.41]

- Can tradeoff sampling speed against filter order
- In high speed converters, making $f_s/f_{sig,max}>10$ is usually impossible or too costly
  - Means that we need fairly high order filters

Classes of Sampling

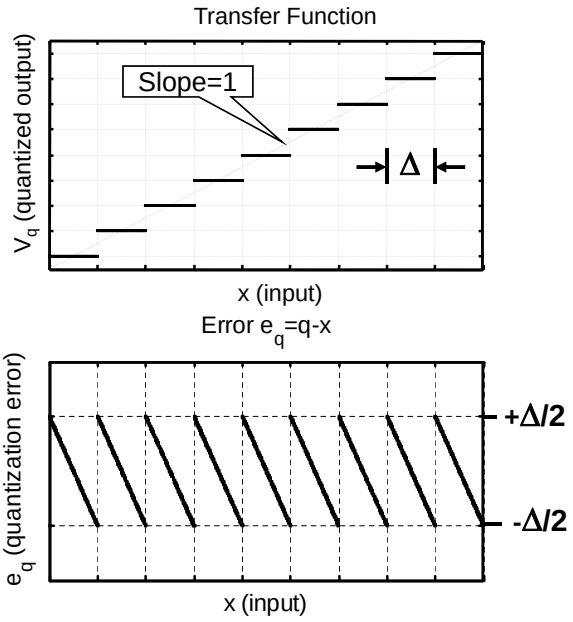
- Nyquist-rate sampling ($f_s > 2 \cdot f_{sig,max}$)
  - Nyquist data converters
  - In practice always slightly oversampled
- Oversampling ($f_s >> 2 \cdot f_{sig,max}$)
  - Oversampled data converters
  - Anti-alias filtering is often trivial
  - Oversampling also helps reduce "quantization noise"
    - More later
- Undersampling, subsampling ($f_s < 2 \cdot f_{sig,max}$)
  - Exploit aliasing to mix RF/IF signals down to baseband
  - See e.g. Pekau & Haslett, JSSC 11/2005

Subsampling



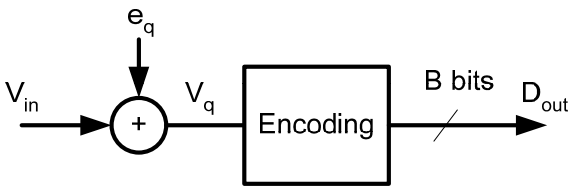
- Aliasing is "non-destructive" if signal is band limited around some carrier frequency
- Downfolding of noise is a severe issue in practical subsampling mixers
  - Typically achieve noise figure no better than 20 dB (!)

Quantization of an Analog Signal



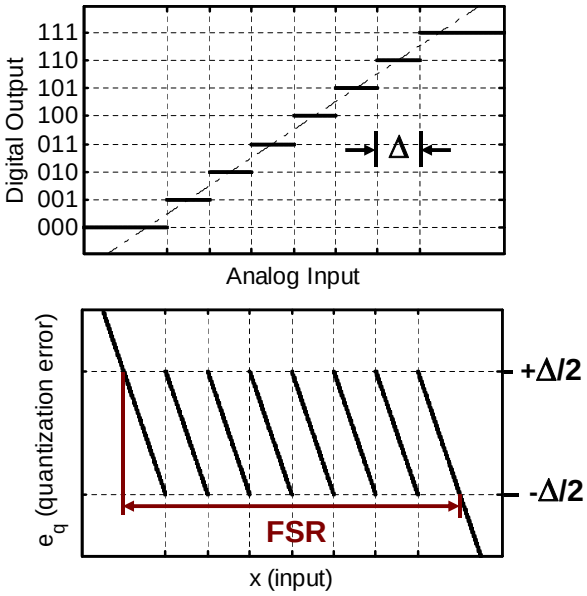
- Quantization step Δ
- Quantization error has sawtooth shape
  - Bounded by –Δ/2, +Δ/2
- Ideally
  - Infinite input range and infinite number of quantization levels
- In practice
  - Finite input range and finite number of quantization levels
  - Output is a digital word (not an analog voltage)

Conceptual Model of a Quantizer



- Encoding block determines how quantized levels are mapped into digital codes
- Note that this model is not meant to represent an actual hardware implementation
  - Its purpose is to show that quantization and encoding are conceptually separate operations
  - Changing the encoding of a quantizer has no interesting implications on its function or performance

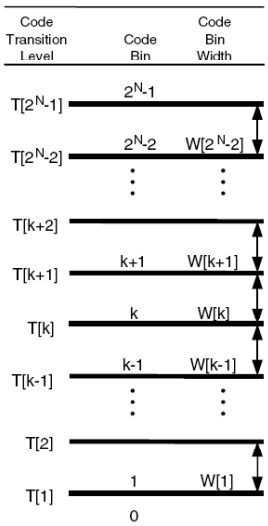
Encoding Example for a B-Bit Quantizer



- Example: B=3
  - 2^3=8 distinct output codes
  - Diagram on the left shows "straight-binary encoding"
  - See e.g. Analog Devices "MT-009: Data Converter Codes" for other encoding schemes
    - http://www.analog.com/en/content/0,2886,760%255F788%255F91285,00.html
- Quantization error grows out of bounds beyond code boundaries
- We define the full scale range (FSR) as the maximum input range that satisfies $|e_q| \leq \Delta/2$
  - Implies that FSR=$2^B \cdot \Delta$

Nomenclature

- **Overloading** - Occurs when an input outside the FSR is applied
- **Transition level** – Input value at the transition between two codes. By standard convention, the transition level T(k) lies between codes k-1 and k
- **Code width** – The difference between adjacent transition levels. By standard convention, the code width W(k)=T(k+1)-T(k)
  - Note that the code width of the first and last code (000 and 111 on previous slide) is undefined
- **LSB size (or width)** – synonymous with code width Δ

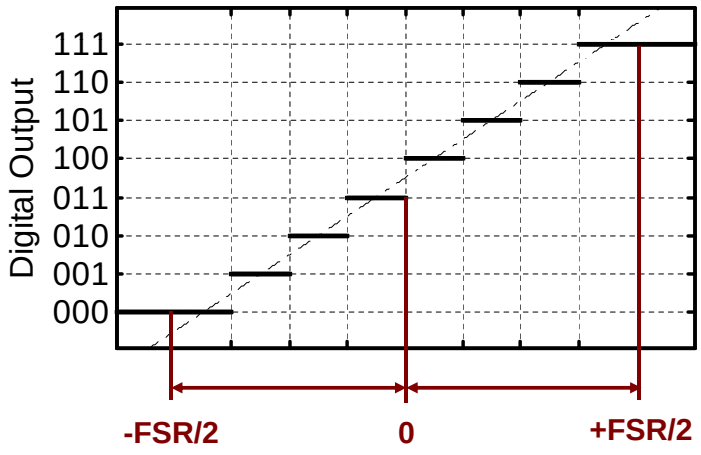


[IEEE Standard 1241-2000]

Implementation Specific Technicalities

- So far, we avoided specifying the absolute location of the code range with respect to "zero" input
- The zero input location depends on the particular implementation of the quantizer
  - Bipolar input, mid-rise or mid-tread quantizer
  - Unipolar input
- The next slide shows the case with
  - Bipolar input
    - The quantizer accepts positive and negative inputs
      - Represents the common case of a differential circuit
  - Mid-rise characteristic
    - The center of the transfer function (zero), coincides with a transition level

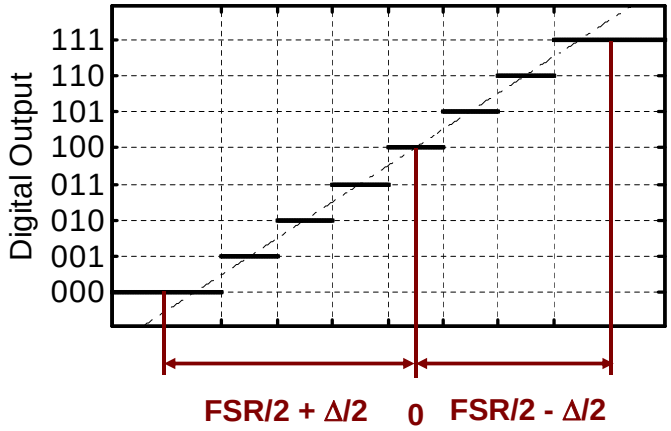
Bipolar Mid-Rise Quantizer



- Nothing new here…

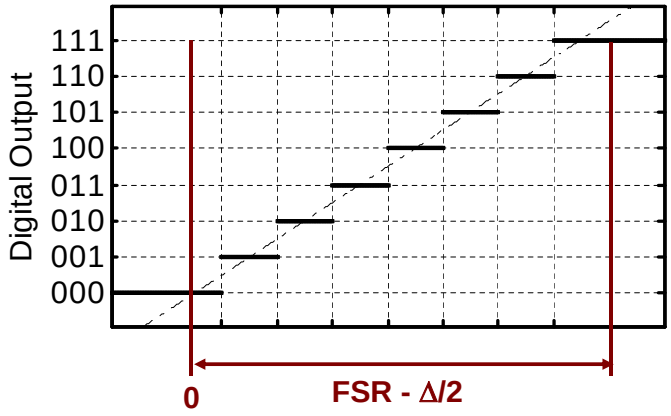
Bipolar Mid-Tread Quantizer

- In theory, less sensitive to infinitesimal disturbance around zero
  - In practice, offsets larger than $\Delta/2$ (due to device mismatch) often make this argument irrelevant
- Asymmetric full-scale range, unless we use odd number of codes



Unipolar Quantizer

- Usually define origin where first code and straight line fit intersect
  - Otherwise, there would be a systematic offset
- Usable range is reduced by $\Delta/2$ below zero

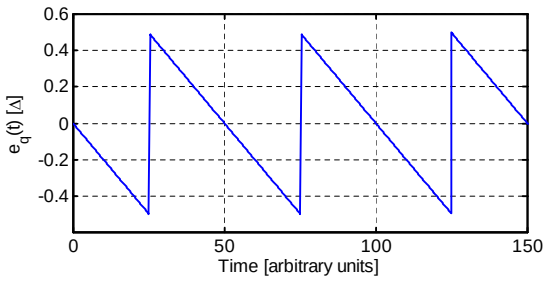


Effect of Quantization Error on Signal

- Two aspects
  - How much noise power does quantization add to samples?
  - How is this noise power distributed in frequency?
- Quantization error is a deterministic function of the signal
  - Should be able answer above questions using a deterministic analysis
  - But, unfortunately, such an analysis strongly depends on the chosen signal and can be very complex
- Strategy
  - Build basic intuition using simple deterministic signals
  - Next, abandon idea of deterministic representation and revert to a "general" statistical model (to be used with caution!)

Ramp Input

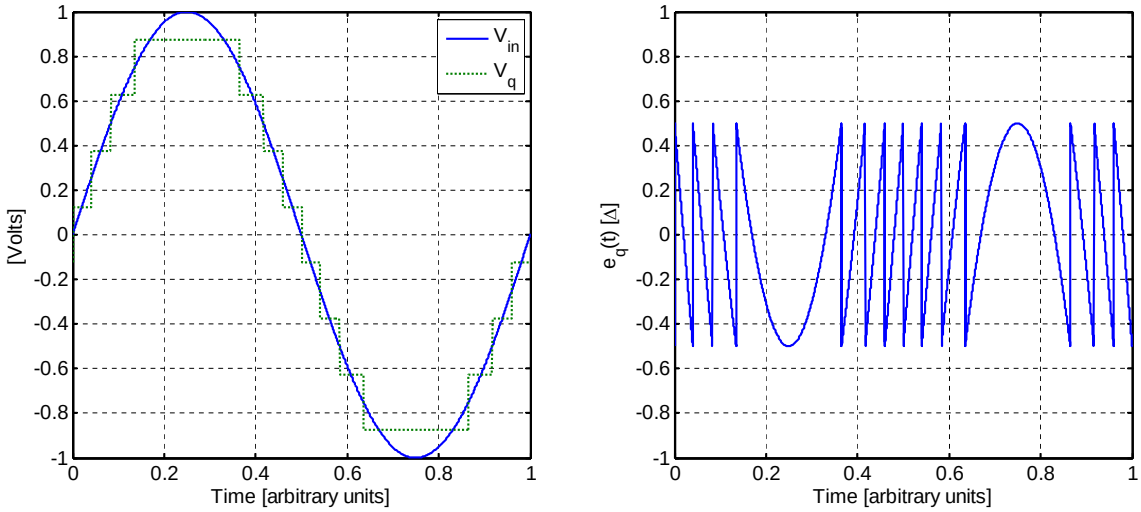
- Applying a ramp signal (periodic sawtooth) at the input of the quantizer gives the following time domain waveform for e_q



- What is the average power of this waveform?
- Integrate over one period

$$\overline{e_q^2} = \frac{1}{T}\int_{-T/2}^{T/2} e_q^2(t)dt \qquad e_q(t) = \frac{\Delta}{T}\cdot t \qquad \boxed{\therefore \overline{e_q^2} = \frac{\Delta^2}{12}}$$

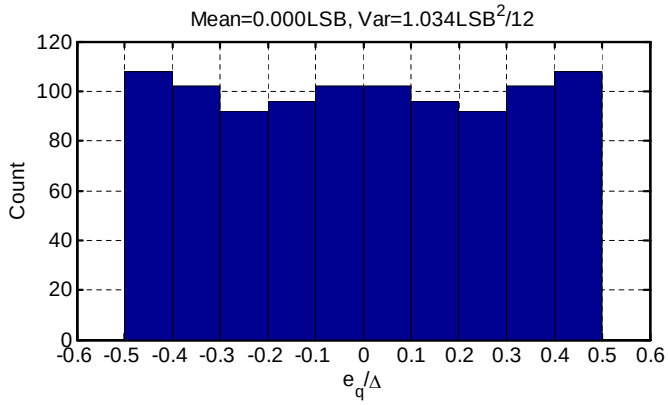
Sine Wave Input



- Integration is not straightforward…

Quantization Error Histogram

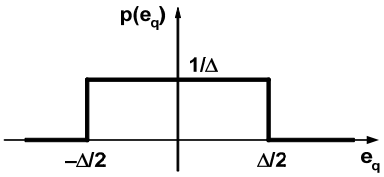
- Sinusoidal input signal with f_{sig}=101Hz, sampled at f_s=1000Hz
- 8-bit quantizer



- Distribution is "almost" uniform
- Can approximate average power by integrating uniform distribution

Statistical Model of Quantization Error

- Assumption: $e_q(x)$ has a uniform probability density
- This approximation holds reasonably well in practice when
  - Signal spans large number of quantization steps
  - Signal is "sufficiently active"
  - Quantizer does not overload

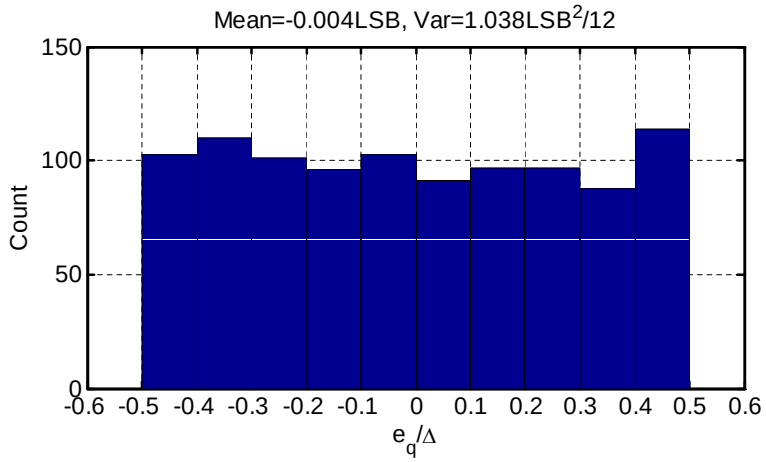


Mean $$\overline{e_q} = \int_{-\Delta/2}^{+\Delta/2} \frac{e_q}{\Delta} de_q = 0$$

Variance $$\overline{e_q^2} = \int_{-\Delta/2}^{+\Delta/2} \frac{e_q^{\,2}}{\Delta} de_q = \frac{\Delta^2}{12}$$

Reality Check (1)

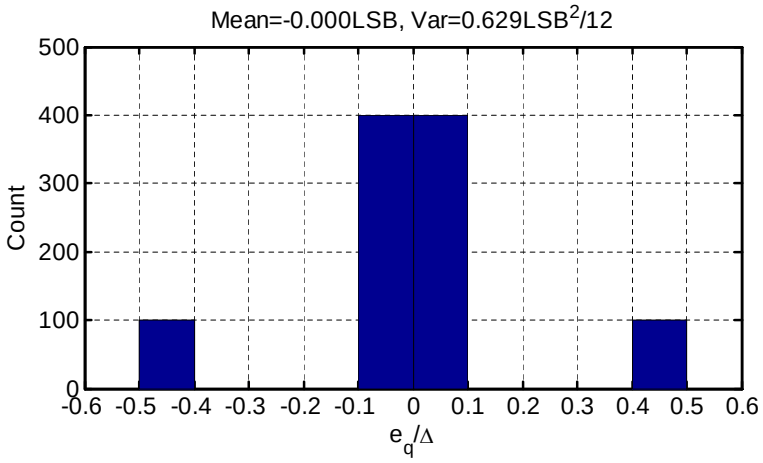
- Input sequence consists of 1000 samples drawn from Gaussian distribution, 4σ=FSR



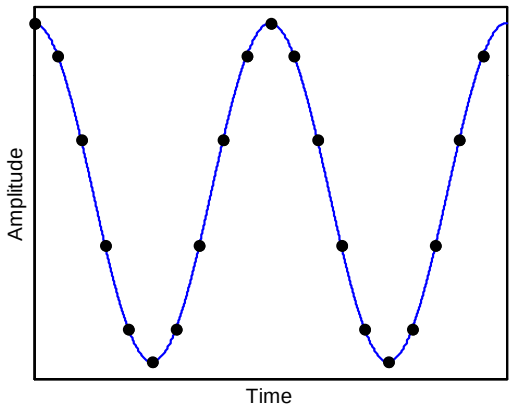
- Error power close to that of uniform approximation

Reality Check (2)

- Another sine wave example, but now $f_{sig}/f_s=100/1000$
- What's going on here?



Analysis (1)



$f_{sig}/f_s=100/1000$

- Sampled signal is repetitive and has only a few distinct values
  - This also means that the quantizer generates only a few distinct values of e_q; not a uniform distribution

Analysis (2)

$$v_{sig}(n) = \cos\left(2\pi \cdot \frac{f_{in}}{f_s} \cdot n\right)$$

- Signal repeats every m samples, where m is the smallest integer that satisfies

$$m \cdot \frac{f_{in}}{f_s} = \text{integer}$$

$$m \cdot \frac{101}{1000} = \text{integer} \quad \Rightarrow \quad m = 1000$$

$$m \cdot \frac{100}{1000} = \text{integer} \quad \Rightarrow \quad m = 10$$

- This means that $e_q(n)$ has at best 10 distinct values, even if we take many more samples

Signal-to-Quantization-Noise Ratio

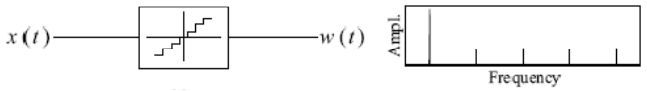
- Assuming uniform distribution of e_q and a full-scale sinusoidal input, we have

$$\text{SQNR} = \frac{P_{sig}}{P_{qnoise}} = \frac{\frac{1}{2}\left(\frac{2^B \Delta}{2}\right)^2}{\frac{\Delta^2}{12}} = 1.5 \times 2^{2B} = 6.02B + 1.76 \text{ dB}$$

B (Number of Bits)	SQNR
8	50 dB
12	74 dB
16	98 dB
20	122 dB

Quantization Noise Spectrum (1)

- How is the quantization noise power distributed in frequency?
  - First think about applying a sine wave to a quantizer, without sampling (output is continuous time)



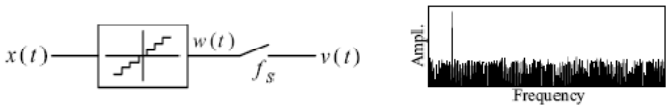
+ many more harmonics

[Y. Tsividis, ICASSP 2004]

- Quantization results in an "infinite" number of harmonics

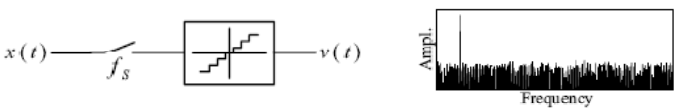
Quantization Noise Spectrum (2)

- Now sample the signal at the output
  - All harmonics (an "infinite" number of them) will alias into band from 0 to $f_s/2$
  - Quantization noise spectrum becomes "white"



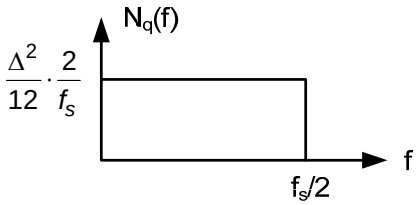
[Y. Tsividis, ICASSP 2004]

- Interchanging sampling and quantization won't change this situation



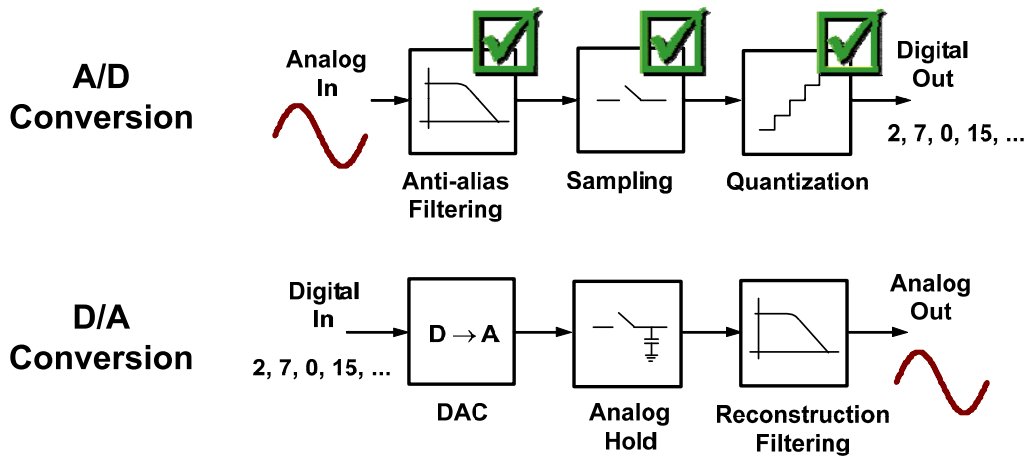
Quantization Noise Spectrum (3)

- Can show that the quantization noise power is indeed distributed (approximately) uniformly in frequency
  - Again, this is provided that the quantization error is "sufficiently random"



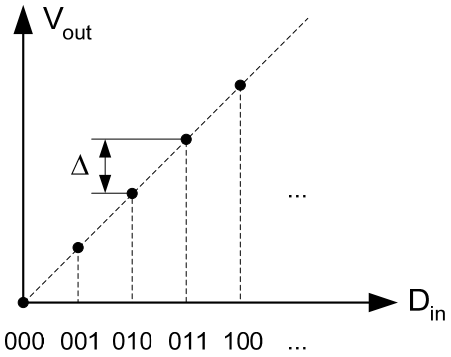
- References
  - W. R. Bennett, "Spectra of quantized signals," Bell Syst. Tech. J., pp. 446-72, July 1948.
  - B. Widrow, "A study of rough amplitude quantization by means of Nyquist sampling theory," IRE Trans. Circuit Theory, vol. CT-3, pp. 266-76, 1956.
  - A. Sripad and D. A. Snyder, "A necessary and sufficient condition for quantization errors to be uniform and white," IEEE Trans. Acoustics, Speech, and Signal Processing, pp. 442-448, Oct 1977.

Recap

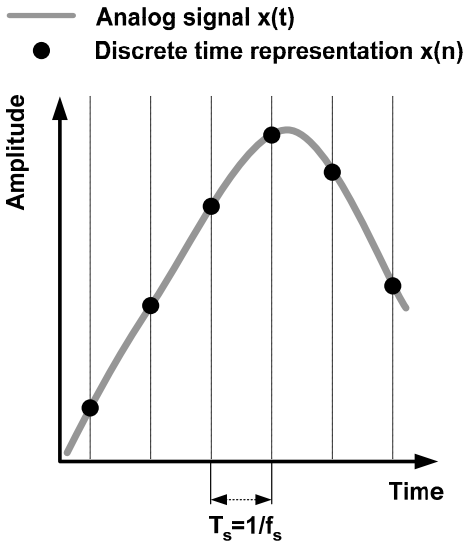


Ideal DAC

- Essentially a digitally controlled voltage, current or charge source
  - Example below is for unipolar DAC
- Ideal DAC does not introduce quantization error!



The Reconstruction Problem



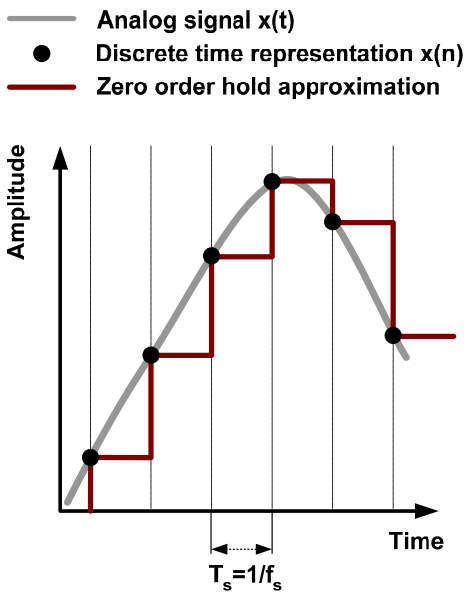
- As long as we sample fast enough, x(n) contains all information about x(t)
  - $f_s > 2 \cdot f_{sig,max}$
- How to reconstruct x(t) from x(n)?
- Ideal interpolation formula

$$x(t) = \sum_{n=-\infty}^{\infty} x(n) \cdot g(t - nT_s)$$

$$g(t) = \frac{sin(\pi f_s t)}{\pi f_s t}$$

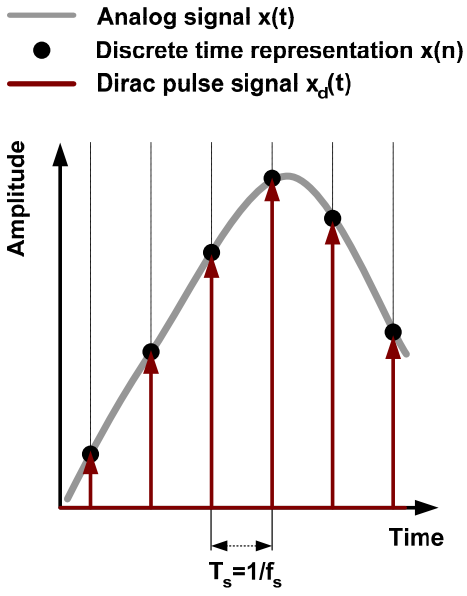
- Very hard to build an analog circuit that does this…

Zero-Order Hold Reconstruction



- The most practical way of reconstructing the continuous time signal is to simply "hold" the discrete time values
  - Either for full period T_s or a fraction thereof
  - Other schemes exist, e.g. “partial-order hold”
    - See [Jha, TCAS II, 11/2008]
- What does this do to the signal spectrum?
- We'll analyze this in two steps
  - First look at infinitely narrow reconstruction pulses

Dirac Pulses



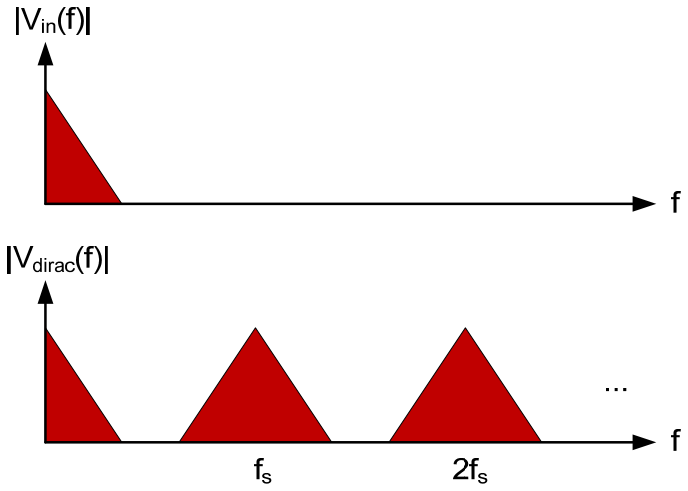
- $x_d(t)$ is zero between pulses
  - Note that x(n) is undefined at these times

$$x_d(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT_s)$$

- Multiplication in time means convolution in frequency
  - Resulting spectrum

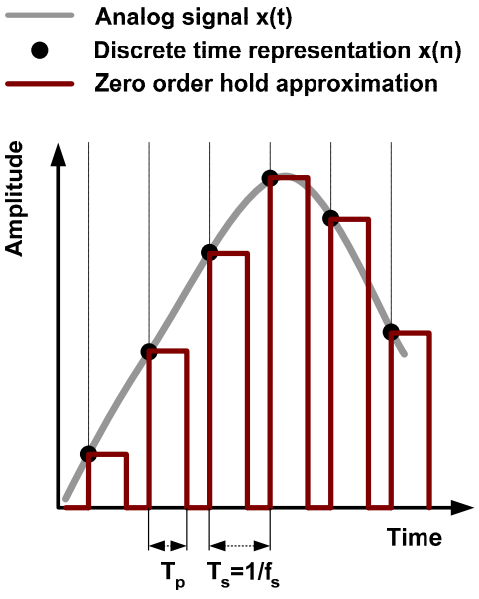
$$X_d(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X\left(f - \frac{n}{T_s}\right)$$

Spectrum



- Spectrum of Dirac signal contains replicas of $V_{in}(f)$ at integer multiples of the sampling frequency

Finite Hold Pulse



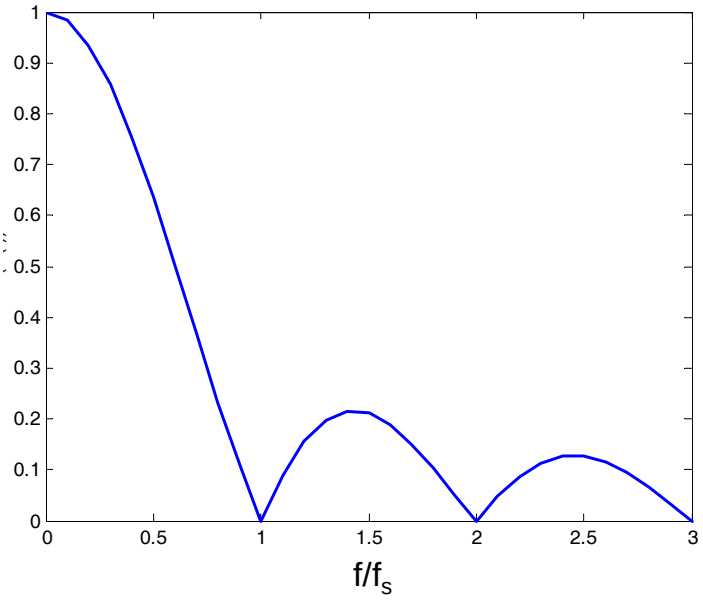
- Consider the general case with a rectangular pulse $0 < T_p \leq T_s$
- The time domain signal on the left follows from convolving the Dirac sequence with a rectangular unit pulse
- Spectrum follows from multiplication with Fourier transform of the pulse

$$H_p(f) = T_p \frac{\sin(\pi f T_p)}{\pi f T_p} \cdot e^{-j\pi f T_p}$$

$$X_p(f) = \underbrace{\frac{T_p}{T_s} \frac{\sin(\pi f T_p)}{\pi f T_p} \cdot e^{-j\pi f T_p}}_{\text{Amplitude Envelope}} \sum_{n=-\infty}^{\infty} X\left(f - \frac{n}{T_s}\right)$$

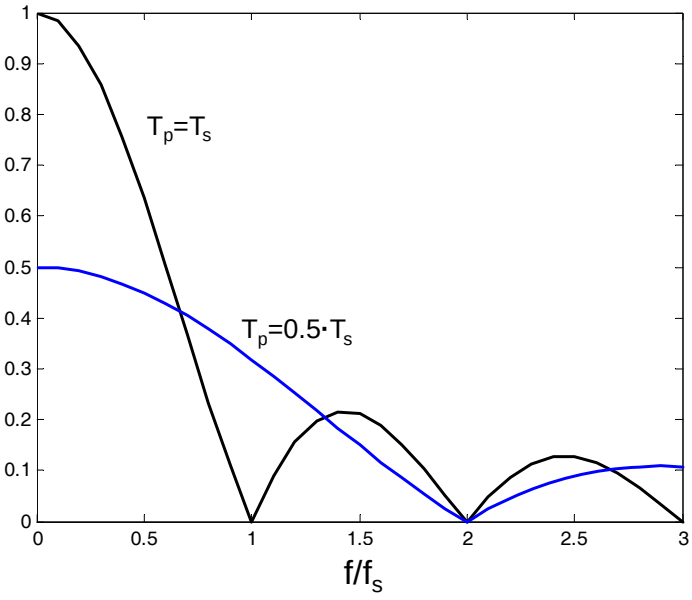
Envelope with Hold Pulse $T_p=T_s$

$$\left|\frac{T_p}{T_s}\frac{\sin(\pi f T_p)}{\pi f T_p}\right|$$

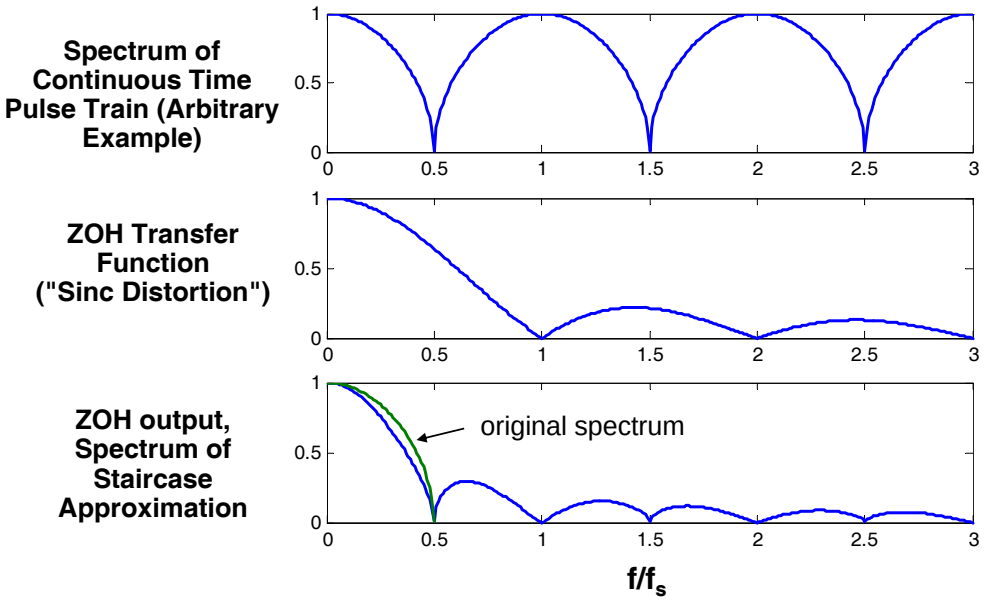


Envelope with Hold Pulse $T_p=0.5\cdot T_s$

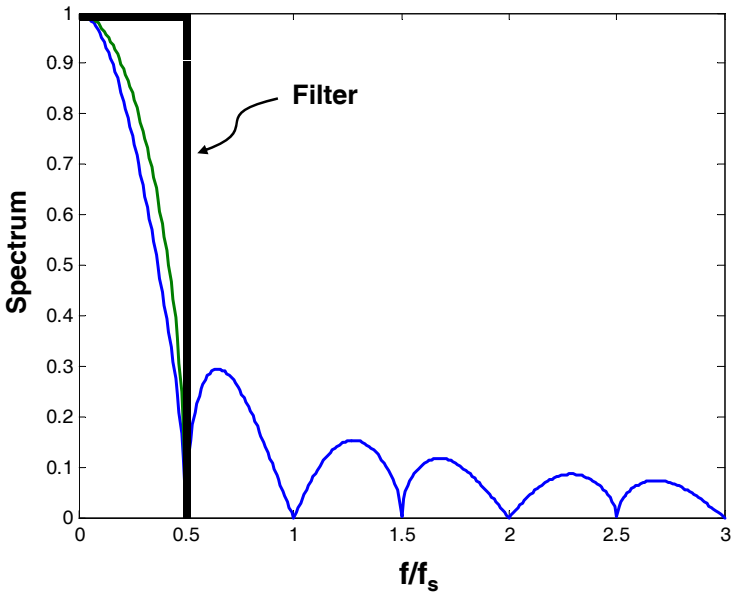
$$\left|\frac{T_p}{T_s}\frac{\sin(\pi f T_p)}{\pi f T_p}\right|$$



Example



Reconstruction Filter



- Also called smoothing filter
- Same situation as with anti-alias filter
  - A brick wall filter would be nice
  - Oversampling helps reduce filter order

Summary

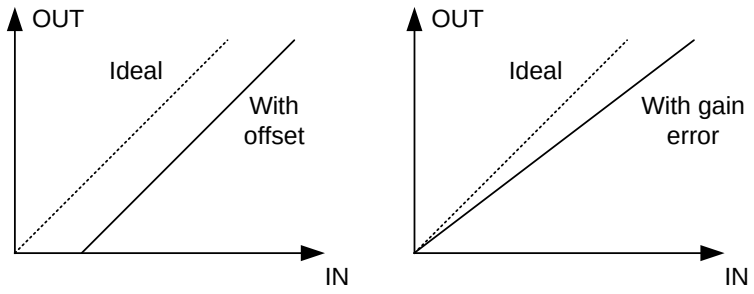
- Must obey sampling theorem $f_s > 2 \cdot f_{sig,max,}$
  - Usually dictates anti-aliasing filter
- If sampling theorem is met, continuous time signal can be recovered from discrete time sequence without loss of information
- A zero order hold in conjunction with a smoothing filter is the most common way to reconstruct
  - May need to add pre- or post-emphasis to cancel droop due to sinc envelope
- Oversampling helps reduce order of anti-aliasing and reconstruction filters

Static Nonidealities

- Static deviations of transfer characteristics from ideality
  - Offset
  - Gain error
  - Differential Nonlinearity (DNL)
  - Integral Nonlinearity (INL)
- Useful references
  - Analog Devices MT-010: The Importance of Data Converter Static Specifications
    - http://www.analog.com/en/content/0,2886,761%255F795%255F91286,00.html
  - "Understanding Data Converters," Texas Instruments Application Report LAA013, 1995.
    - http://focus.ti.com/lit/an/slaa013/slaa013.pdf

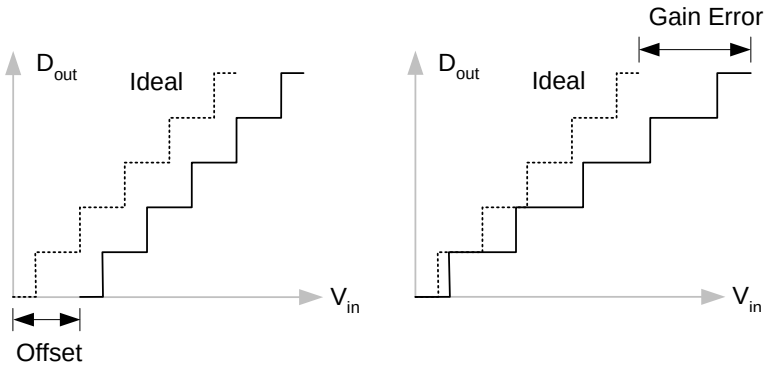
Offset and Gain Error

- Conceptually simple, but lots of (uninteresting) subtleties in how exactly these errors should be defined
  - Unipolar versus bipolar, endpoint versus midpoint specification
  - Definition in presence of nonlinearities
- General idea (neglecting staircase nature of transfer functions):



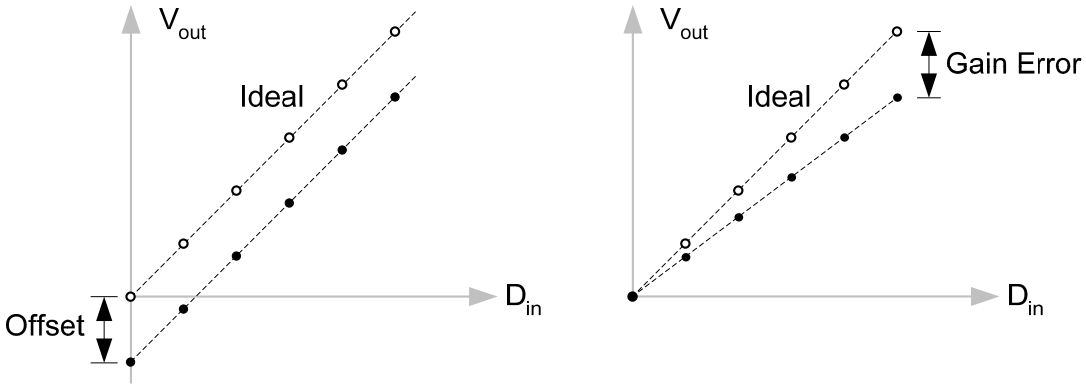
ADC Offset and Gain Error

- Definitions based on bottom and top endpoints of transfer characteristic
  - ½ LSB before first transition and ½ LSB after last transition
  - Offset is the deviation of bottom endpoint from its ideal location
  - Gain error is the deviation of top endpoint from its ideal location with offset removed
- Both quantities are measured in LSB or as percentage of full-scale range



DAC Offset and Gain Error

- Same idea, except that endpoints are directly defined by analog output values at minimum and maximum digital input
- Also note that errors are specified along the vertical axis



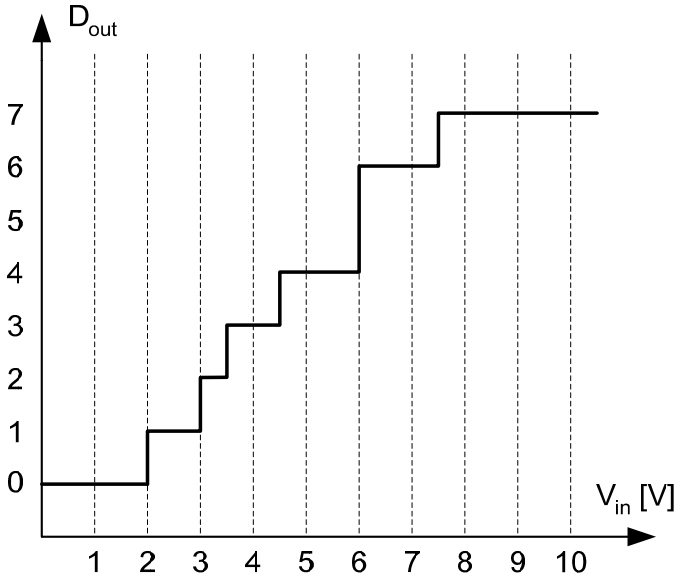
Comments on Offset and Gain Errors

- Definitions on the previous slides are the ones typically used in industry
  - IEEE Standard suggest somewhat more sophisticated definitions based on least square curve fitting
    - Technically more suitable metric when the transfer characteristics are significantly non-uniform or nonlinear
- Generally, it is non-trivial to build a converter with very good gain/offset specifications
  - Nevertheless, since gain and offset affect all codes uniformly, these errors tend to be easy to correct
    - E.g. using a digital pre- or post-processing operation
  - Also, many applications are insensitive to a certain level of gain and offset errors
    - E.g. audio signals, communication-type signals, ...
- More interesting aspect: linearity
  - DNL and INL

Differential Nonlinearity (DNL)

- In an ideal world, all ADC codes would have equal width; all DAC output increments would have same size
- DNL(k) is a vector that quantifies for each code k the deviation of this width from the "average" width (step size)
- DNL(k) is a measure of uniformity, it does not depend on gain and offset errors
  - Scaling and shifting a transfer characteristic does not alter its uniformity and hence DNL(k)
- Let's look at an example

ADC DNL Example (1)



Code (k)	W [V]
0	undefined
1	1
2	0.5
3	1
4	1.5
5	0
6	1.5
7	undefined

ADC DNL Example (2)

- What is the average code width?
  - ADC with perfect uniformity would divide the range between first and last transition into 6 equal pieces
  - Hence calculate average code width (i.e. LSB size) as

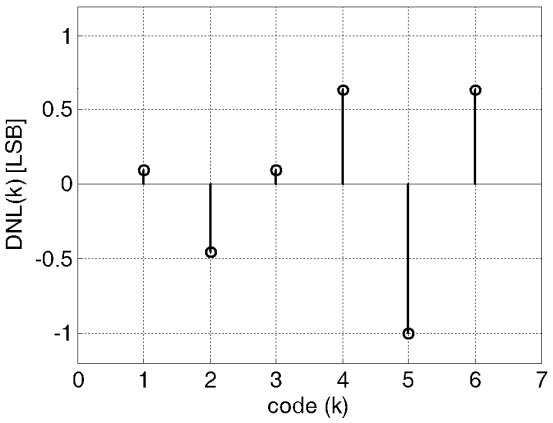
$$W_{avg} = \frac{7.5V - 2V}{6} = 0.9167V$$

- Now calculate DNL(k) for each code k using

$$DNL(k) = \frac{W(k) - W_{avg}}{W_{avg}}$$

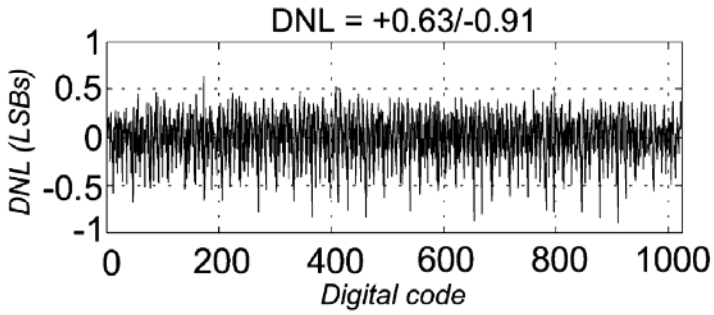
Result

Code (k)	DNL [LSB]
1	0.09
2	-0.45
3	0.09
4	0.64
5	-1.00
6	0.64



- Positive/negative DNL implies wide/narrow code, respectively
- DNL = -1 LSB implies missing code
- Impossible to have DNL < -1 LSB for an ADC
  - But possible to have DNL > +1 LSB
- Can show that sum over all DNL(k) is equal to zero

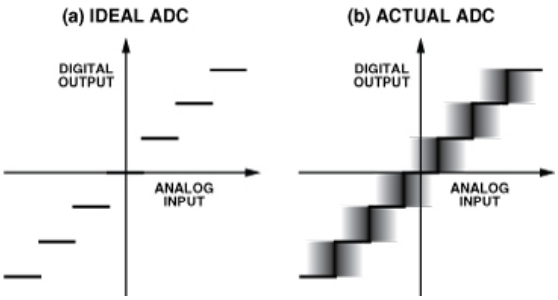
A Typical ADC DNL Plot



[Ahmed, JSSC 12/2005]

- People speak about DNL often only in terms of min/max number across all codes
  - E.g. DNL = +0.63/-0.91 LSB
- Might argue in some cases that any code with DNL < -0.9 LSB is essentially a missing code
  - Why ?

Impact of Noise



[W. Kester, "ADC Input Noise: The Good, The Bad, and The Ugly. Is No Noise Good Noise?" Analogue Dialogue, Feb. 2006]

- In essentially all moderate to high-resolution ADCs, the transition levels carry noise that is somewhat comparable to the size of an LSB
  - Noise "smears out" DNL, can hide missing codes
- Especially for converters whose input referred (thermal) noise is larger than an LSB, DNL is a "fairly useless" metric

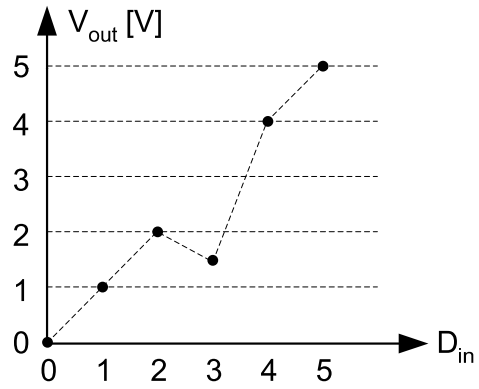
DAC DNL

- Same idea applies
  - Find output increments for each digital code
  - Find increment that divides range into equal steps
  - Calculate DNL for each code k using

$$\text{DNL(k)} = \frac{\text{Step(k)} - \text{Step}_{\text{avg}}}{\text{Step}_{\text{avg}}}$$

- One difference between ADC and DAC is that DAC DNL can be less than -1 LSB
  - How ?

Non-Monotonic DAC

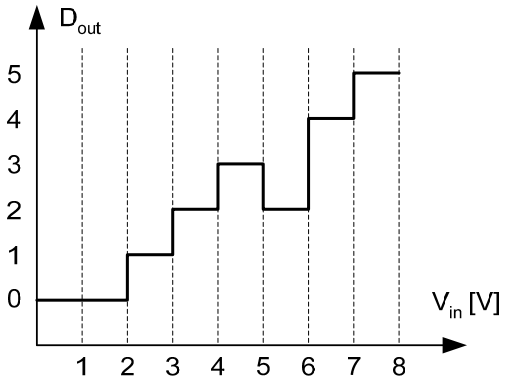


$$\text{DNL(3)} = \frac{\text{Step(3)} - \text{Step}_{\text{avg}}}{\text{Step}_{\text{avg}}}$$

$$= \frac{-0.5\text{V} - 1\text{V}}{1\text{V}} = -1.5\text{LSB}$$

- In a DAC, DNL < -1LSB implies non-monotinicity
- How about a non-monotonic ADC?

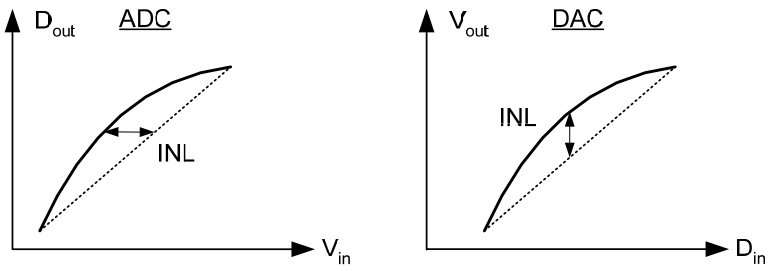
Non-Monotonic ADC



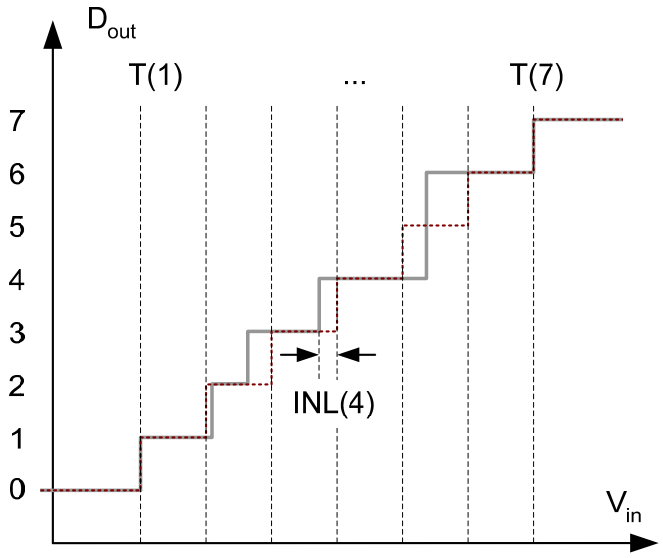
- Code 2 has two transition levels $\Rightarrow$ W(2) is ill defined
  - DNL is ill-defined!
- Not a very big issue, because a non-monotonic ADC is usually not what we'll design for in practice…

Integral Nonlinearity (INL)

- General idea
  - For each "relevant point" of the transfer characteristic, quantify distance from a straight line drawn through the endpoints
    - An alternative, less common definition uses a least square fit line as a reference
  - Just as with DNL, the INL of a converter is by definition independent of gain and offset errors



ADC INL Example (1)



- "Straight line" reference is uniform staircase between first and last transition
- INL for each code is

$$INL(k) = \frac{T(k) - T_{uniform}(k)}{W_{avg}}$$

- Obviously INL(1) = 0 and INL(7) = 0
- INL(0) is undefined

ADC INL Example (2)

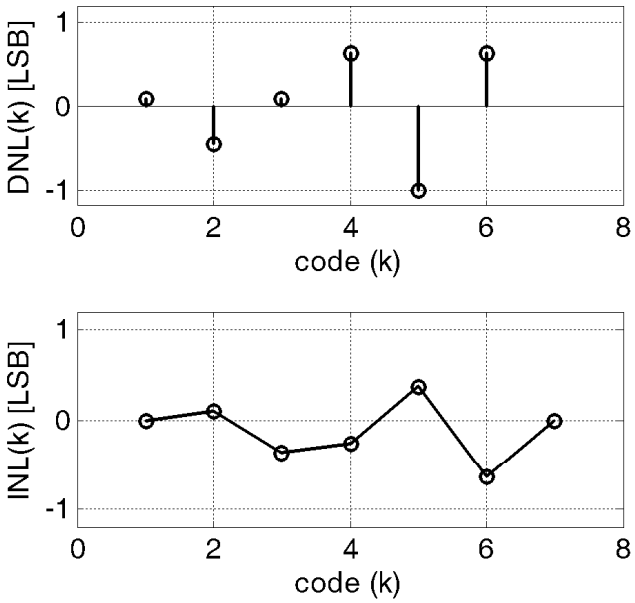
- Can show that

$$INL(k) = \sum_{i=1}^{k-1} DNL(i)$$

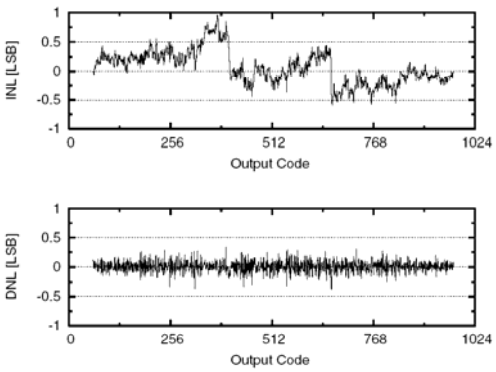
- Means that once we computed DNL, we can easily find INL using a cumulative sum operation on the DNL vector
- Using DNL values from last lecture, we find

Code (k)	DNL [LSB]	INL [LSB]
1	0.09	0
2	-0.45	0.09
3	0.09	-0.36
4	0.64	-0.27
5	-1.00	0.36
6	0.64	-0.64
7	undefined	0

Result



A Typical ADC DNL/INL Plot



[Ishii, Custom Integrated Circuits Conference, 2005]

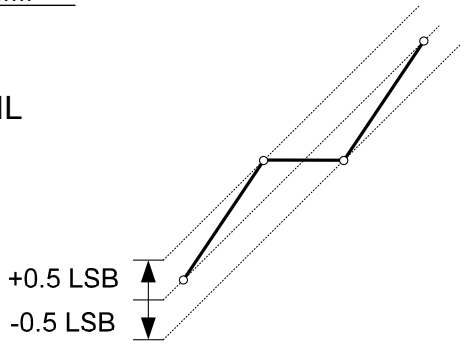
- DNL/INL signature often reveals architectural details
  - E.g. major transitions
  - We'll see more examples in the context of DACs
- Since INL is a cumulative measure, it turns out to be less sensitive than DNL to thermal noise "smearing"

DAC INL

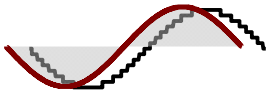
- Same idea applies
  - Find ideal output values that lie on a straight line between endpoints
  - Calculate INL for each code k using

$$INL(k) = \frac{V_{out}(k) - V_{outuniform}(k)}{Step_{avg}}$$

- Interesting property related to DAC INL
  - If for all codes |INL| < 0.5 LSB, it follows that all |DNL| < 1 LSB
  - A sufficient (but not necessary) condition for monotonicity



Spectral Performance Metrics



Katelijn Vleugels
Stanford University

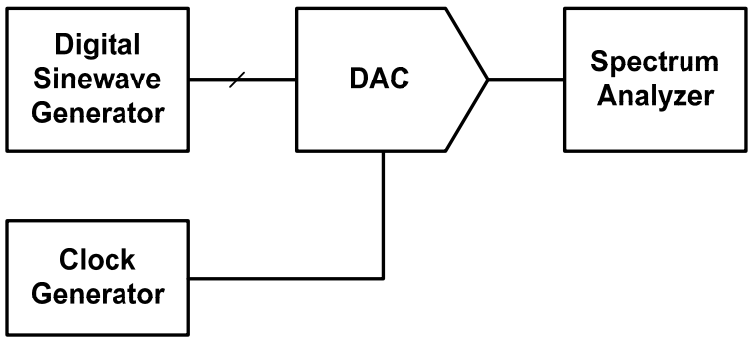
Spectral Performance Metrics

- Spectral performance metrics follow from looking at converter or building block output spectrum
- Basic idea:
  - Apply one or more tones at converter input
  - Expect same tone(s) at output, all other frequency components represent non-idealities
- Important to realize that both static and dynamic errors contribute to frequency domain non-ideality
  - Static: DNL,INL
  - Dynamic: glitch impulse, aperture uncertainty, settling time, …
    - We'll look at these later, in the context of specific circuits

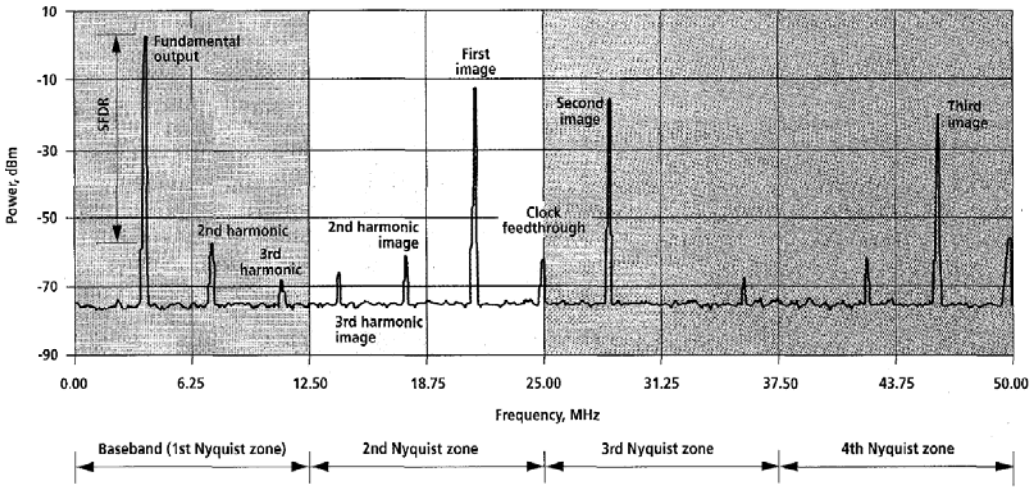
Alphabet Soup of Spectral Metrics

- SNR - Signal-to-noise ratio
- SNDR (SINAD) - Signal-to-(noise+distortion) ratio
- ENOB - Effective number of bits
- DR - Dynamic range
- SFDR - Spurious free dynamic range
- THD - Total harmonic distortion
- ERBW - Effective Resolution Bandwidth
- IMD - Intermodulation distortion
- MTPR - Multi-tone power ratio

DAC Tone Test/Simulation

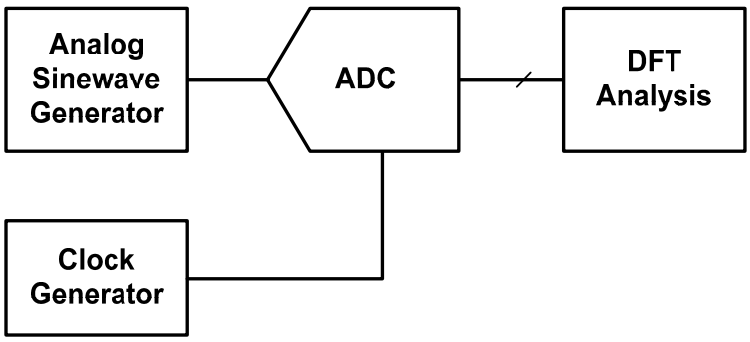


Typical DAC Output Spectrum



[Hendriks, "Specifying Communications DACs, IEEE Spectrum, July 1997]

ADC Tone Test/Simulation



Discrete Fourier Transform Basics

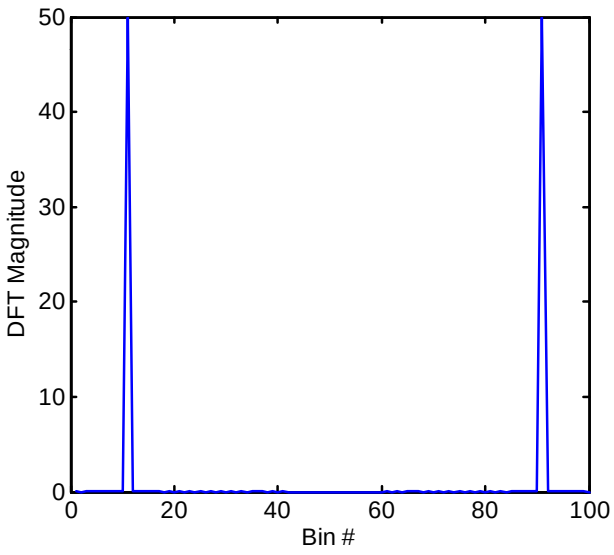
- DFT takes a block of N time domain samples (spaced $T_s=1/f_s$) and yields a set of N frequency bins

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$$

- Bin k represents frequency content at $k \cdot f_s/N$ [Hz]
- DFT frequency resolution
  - Proportional to $1/(N \cdot T_s)$ in [Hz/bin]
  - $N \cdot T_s$ is total time spent gathering samples
- A DFT with $N=2^{integer}$ can be found using a computationally efficient algorithm
  - FFT = Fast Fourier Transform

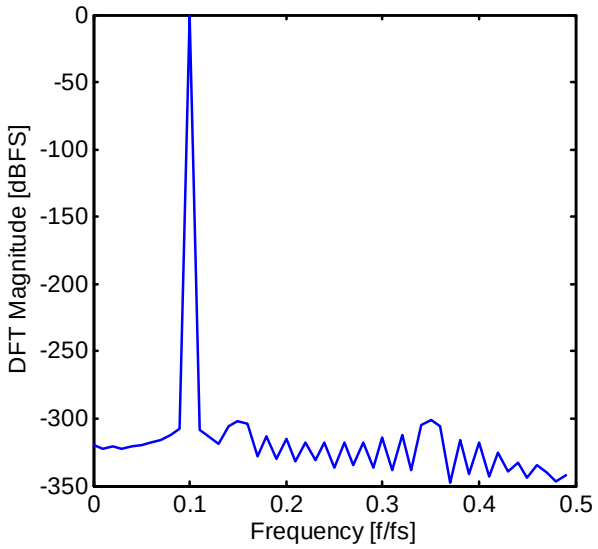
Matlab Example

```
clear;
N  = 100;
fs = 1000;
fx = 100;

x = cos(2*pi*fx/fs*[0:N-1]);
s = abs(fft(x));
plot(s, 'linewidth', 2);
```



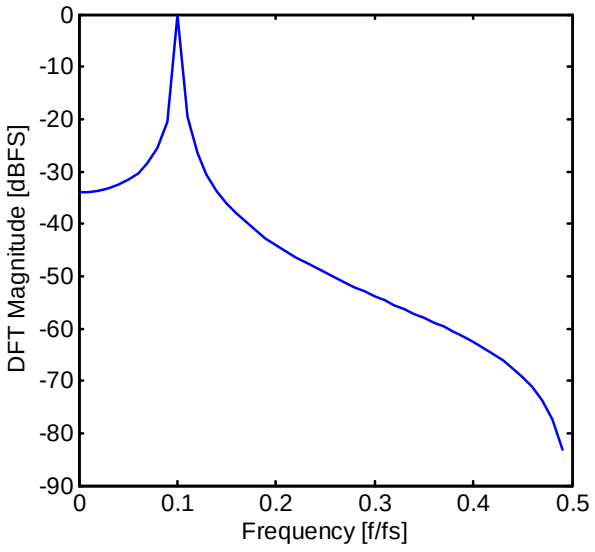
Normalized Plot with Frequency Axis

```
N  = 100;
fs = 1000;
fx = 100;
FS = 1; % full-scale amplitude

x = FS*cos(2*pi*fx/fs*[0:N-1]);
s = abs(fft(x));
% remove redundant half of spectrum
s = s(1:end/2);
% normalize magnitudes to dBFS
% dbFS = dB relative to full-scale
s = 20*log10(2*s/N/FS);
% frequency vector
f = [0:N/2-1]/N;

plot(f, s, 'linewidth', 2);
xlabel('Frequency [f/fs]')
ylabel('DFT Magnitude [dBFS]')
```



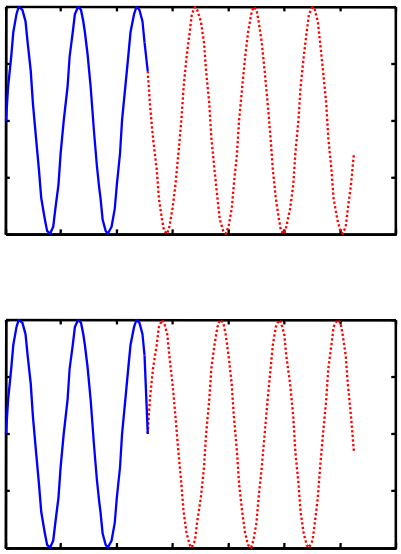
Another Example

- Same as before, but now f_x=101
- This doesn't look the spectrum of a sinusoid…
- What's going on?



Spectral Leakage

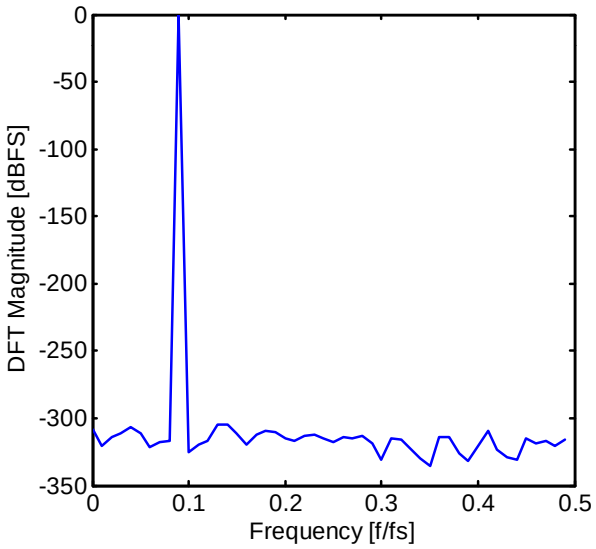
- DFT implicitly assumes that data repeats every N samples
- A sequence that contains a non-integer number of sine wave cycles has discontinuities in its periodic repetition
  - Discontinuity looks like a high frequency signal component
  - Power spreads across spectrum
- Two ways to deal with this
  - Ensure integer number of periods
  - Windowing



Integer Number of Cycles

```
N  = 100;
cycles = 9;
fs = 1000;
fx = fs*cycles/N;
```

- Usable test frequencies are limited to a multiple of f_s/N

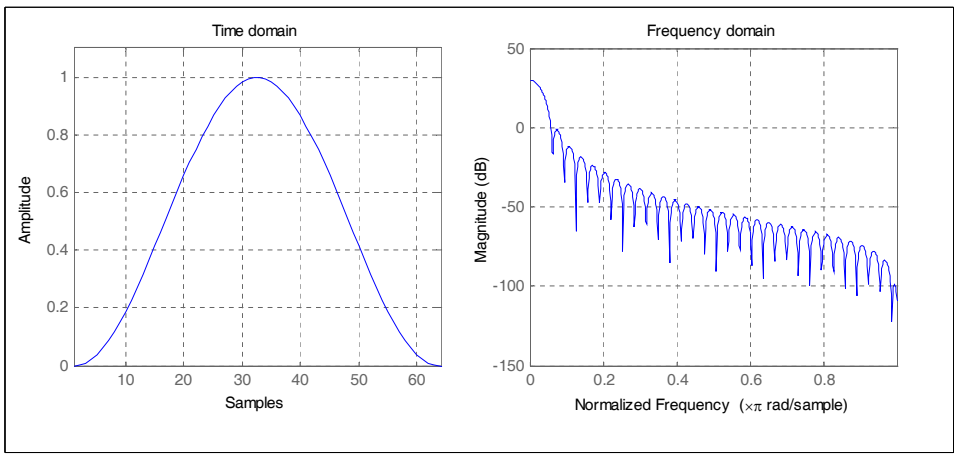


Windowing

- Spectral leakage can be attenuated by windowing the time samples prior to the DFT
- Windows taper smoothly down to zero at the beginning and the end of the observation window
- Time domain samples are multiplied by window coefficients on a sample-by-sample basis
  - Means convolution in frequency
  - Sine wave tone and other spectral components smear out over several bins
- Lots of window functions to chose from
  - Tradeoff: attenuation versus smearing
- Example: Hann Window

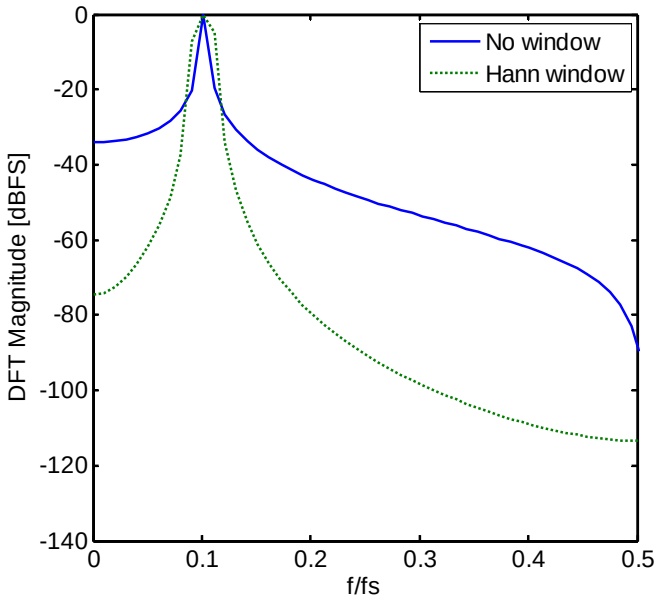
Hann Window

```
N=64;
wvtool(hann(N))
```



Spectrum with Window

```
N  = 100;
fs = 1000;
fx = 101;
A  = 1;

x = A*cos(2*pi*fx/fs*[0:N-1]);
s  = abs(fft(x));
x1 = x.*hann(N);
s1 = abs(fft(x1));
```



Integer Cycles versus Windowing

- Integer number of cycles
  - Test signal falls into single DFT bin
  - Requires careful choice of signal frequency
  - Ideal for simulations
  - In lab measurements, can lock sampling and signal frequency generators (PLL)
    - "Coherent sampling"
- Windowing
  - No restrictions on signal frequency
  - Signal and harmonics distributed over several DFT bins
    - Beware of smeared out nonidealities…
  - Requires more samples for given accuracy
- More info
  - http://www.maxim-ic.com/appnotes.cfm/appnote_number/1040

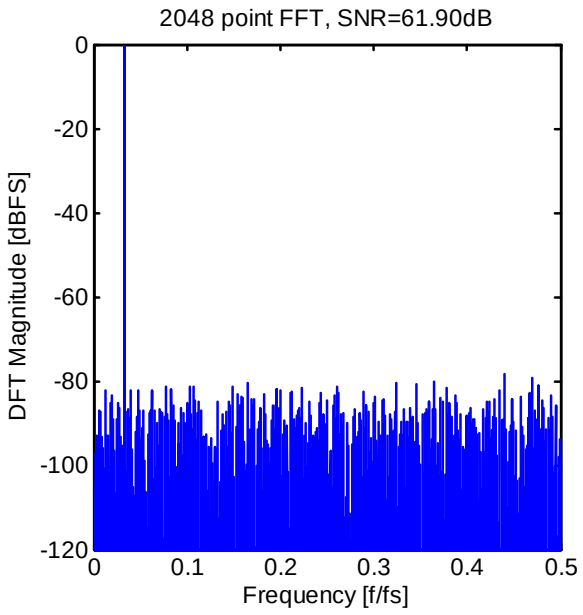
Example

- Now that we've "calibrated" our test system, let's look at some spectra that involve nonidealities
- First look at quantization noise introduced by an ideal quantizer

```
N  = 2048;
cycles = 67;
fs = 1000;
fx = fs*cycles/N;
LSB = 2/2^10;

%generate signal, quantize (mid-tread) and take FFT
x = cos(2*pi*fx/fs*[0:N-1]);
x = round(x/LSB)*LSB;
s  = abs(fft(x));
s = s(1:end/2)/N*2;

% calculate SNR
sigbin = 1 + cycles;
noise = [s(1:sigbin-1), s(sigbin+1:end)];
snr = 10*log10( s(sigbin)^2/sum(noise.^2) );
```


Spectrum with Quantization Noise

- Spectrum looks fairly uniform
- Signal-to-quantization noise ratio is given by power in signal bin, divided by sum of all noise bins



SQNR

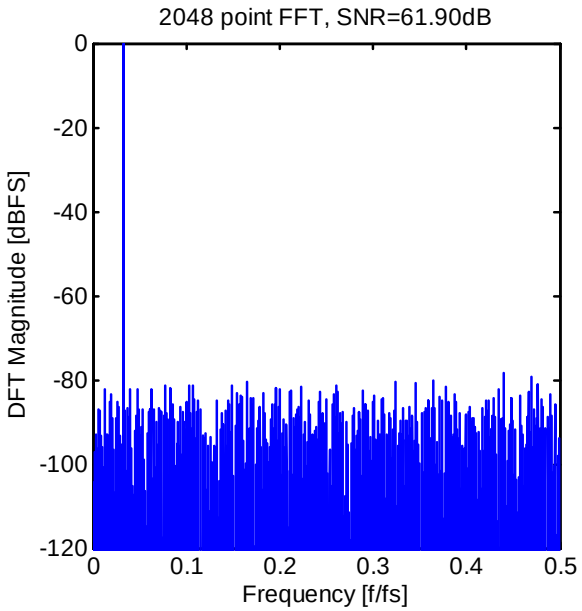
$$\text{SQNR} = \frac{\text{Signal Power}}{\text{Quantization Noise Power}}$$

$$= \frac{\frac{1}{2}\left(\frac{V_{FS}}{2}\right)^2}{\frac{1}{12}\left(\frac{V_{FS}}{2^N}\right)^2} = \frac{3}{2}\cdot 2^{2N}$$

$$= 6.02\cdot N + 1.76 \text{ [dB]}$$

$$= 6.02\cdot 10 + 1.76 \text{ [dB]}$$

$$= 61.9 \text{ dB}$$



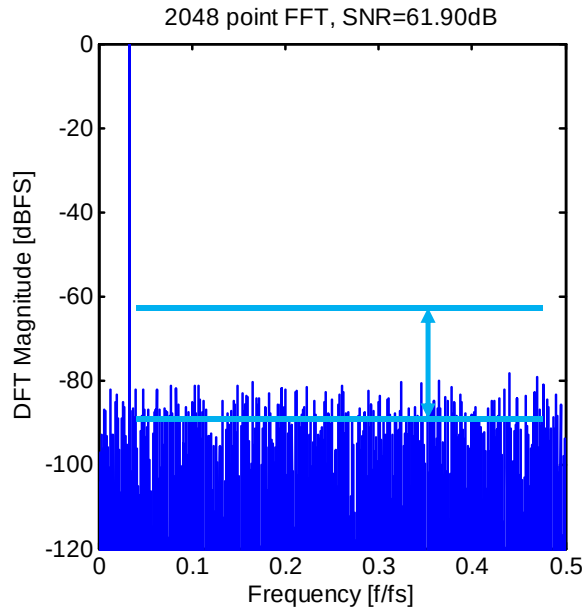
FFT Noise Floor

$$N_{floor} = -61.9 \text{ dBc} - 10\log\left(\frac{2048}{2}\right)$$

$$= -61.9 \text{ dBc} - 30.1 \text{ dB}$$

$$= -92 \text{ dBc}$$

- Depends on FFT size
- Plot is “useless” if FFT size is not specified

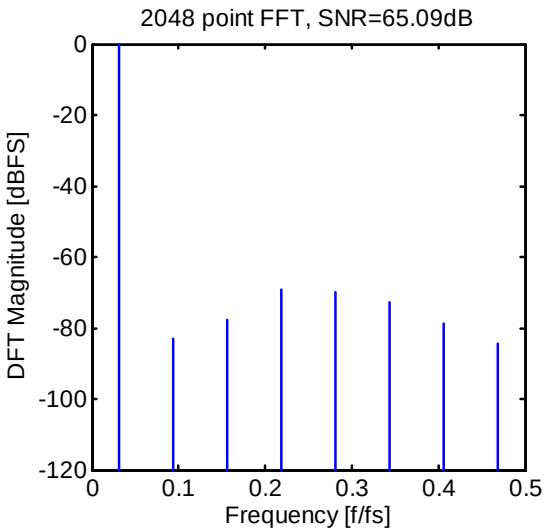


DFT Plot Annotation

- DFT plots are fairly meaningless unless you clearly specifiy the underlying conditions
- Most common annotation
  - Specify how many DFT points were used (N)
- Less common options
  - Shift DFT noise floor by $10\log_{10}(N/2)$dB
  - Normalize with respect to bin width in Hz and express noise as power spectral density
    - "Noise power in 1 Hz bandwidth"

Periodic Quantization Noise

- Same as before, but cycles = 64 (instead of 67)
- $f_x = f_s \cdot 64/2048 = f_s/32$
- Quantization noise is highly deterministic and periodic
- For more random and "white" quantizion noise, it is best to make N and cycles mutually prime
  - GCD(N,cycles)=1

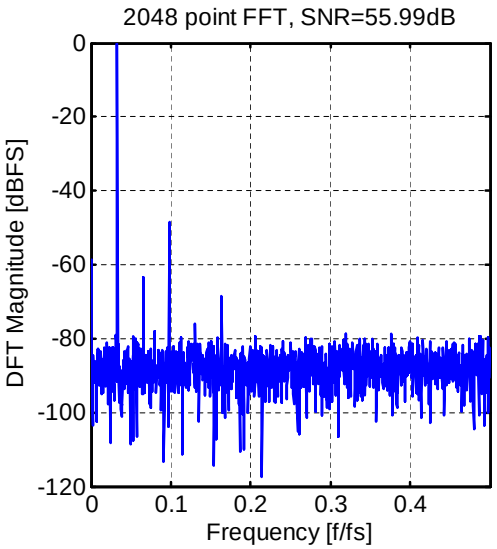


Typical ADC Output Spectrum

- Fairly uniform noise floor due to additional electronic noise
- Harmonics due to nonlinearities
- Definition of SNR

$$SNR = \frac{\text{Signal Power}}{\text{Total Noise Power}}$$

- Total noise power includes all bins except DC, signal, and 2nd through 7th harmonic
  - Both quantization noise and electronic noise affect SNR



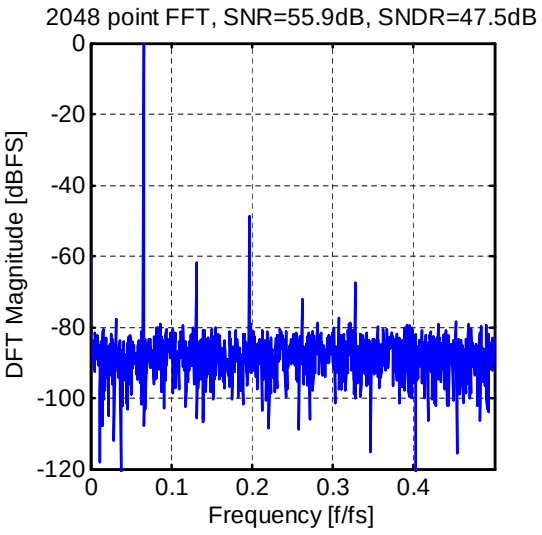
SNDR and ENOB

- Definition

$$SNDR = \frac{\text{Signal Power}}{\text{Noise and Distortion Power}}$$

- Noise and distortion power includes all bins except DC and signal
- Effective number of bits

$$ENOB = \frac{\text{SNDR(dB)-1.76dB}}{\text{6.02dB}}$$



Effective Number of Bits

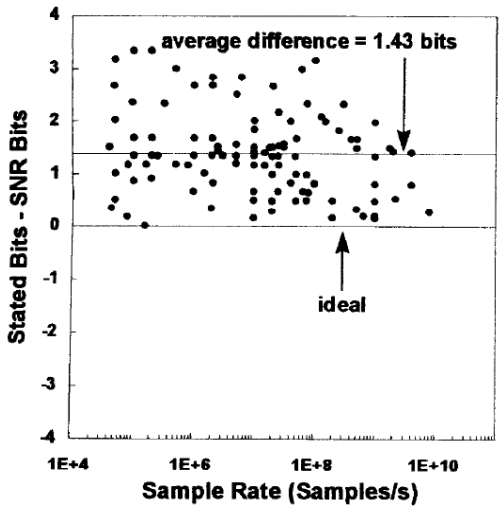
- Is a 10-Bit converter with 47.5dB SNDR really a 10-bit converter?

$$\text{ENOB} = \frac{47.5\text{dB} - 1.76\text{dB}}{6.02\text{dB}} = 7.6$$

- We get ideal ENOB only for zero electronic noise, perfect transfer function with zero INL, ...
- Low electronic noise is costly
  - Cutting thermal noise down by 2x, can cost 4x in power dissipation
- Rule of thumb for good power efficiency: ENOB < B-1
  - B is the "number of wires" coming out of the ADC or the so called "stated resolution"

Survey Data

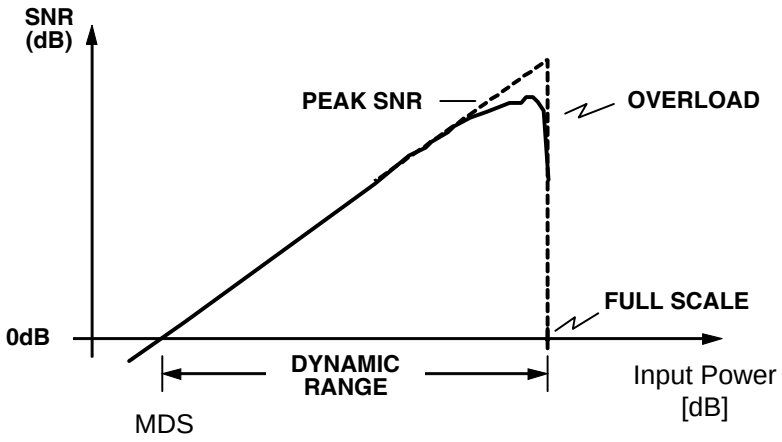
$$\text{SNRBits} = \frac{\text{SNR(dB)-1.76dB}}{6.02\text{dB}}$$



R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE J. on Selected Areas in Communications*, pp. 539-50, April 1999

Dynamic Range

$$DR = \frac{\text{Maximum Signal Power}}{\text{Minimum Detectable Signal}} \geq SNR_{peak}$$

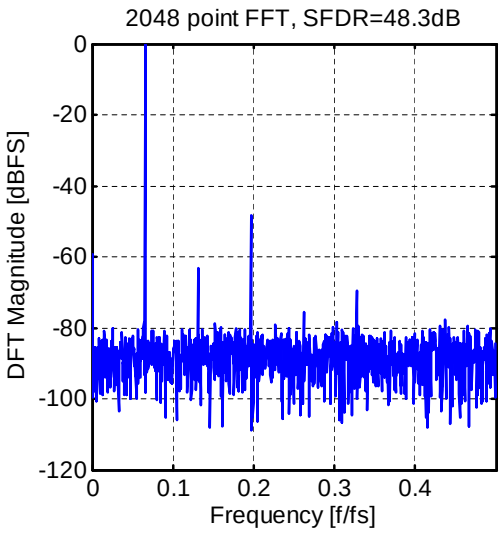


SFDR

- Definition of "Spurious Free Dynamic Range"

$$SFDR = \frac{\text{Signal Power}}{\text{Largest Spurious Power}}$$

- Largest spur is often (but not necessarily) a harmonic of the input tone



SDR and THD

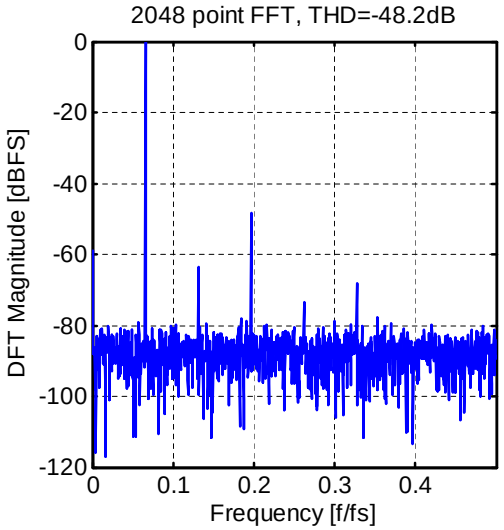
- Signal-to-distortion ratio

$$\text{SDR} = \frac{\text{Signal Power}}{\text{Total Distortion Power}}$$

- Total harmonic distortion

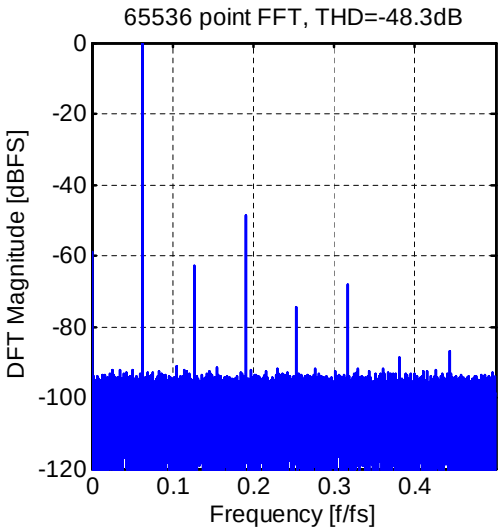
$$\text{THD} = \frac{\text{Total Distortion Power}}{\text{Signal Power}} = \frac{1}{\text{SDR}}$$

- By convention, total distortion power consists of 2nd through 7th harmonic
- Is there a 6th and 7th harmonic in the plot to the right?



Lowering the Noise Floor

- Increasing the FFT size let's us lower the noise floor and reveal low level harmonics



Aliasing

- Harmonics can appear at "arbitrary" frequencies due to aliasing

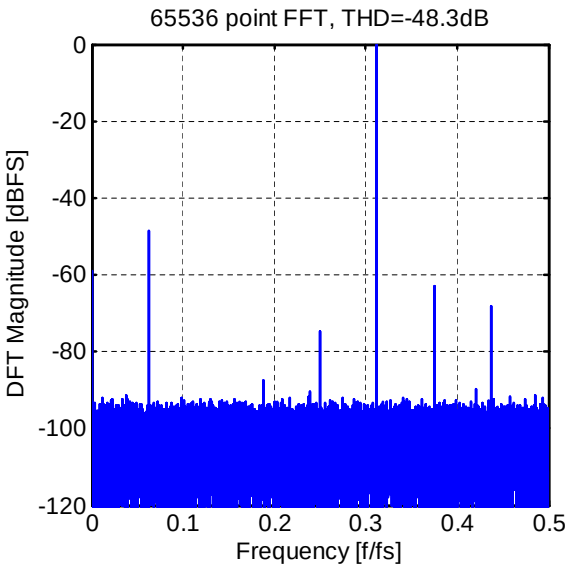
$f_1 = f_x = 0.3125\ f_s$

$f_2 = 2\ f_1 = 0.6250\ f_s \rightarrow 0.3750\ f_s$

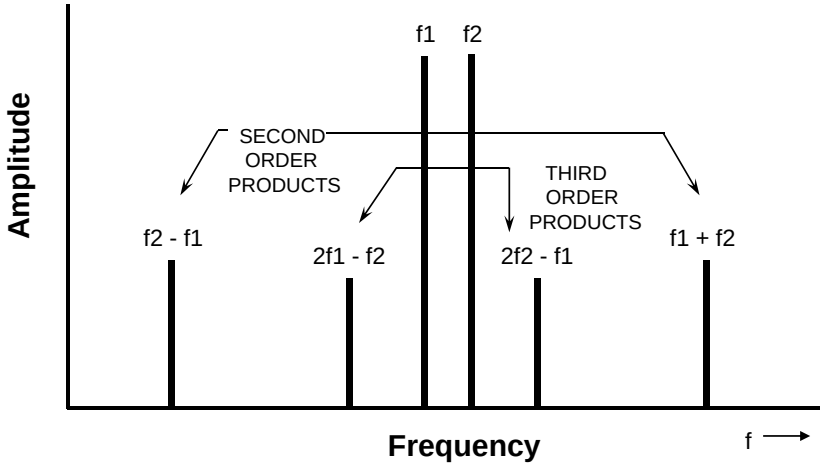
$f_3 = 3\ f_1 = 0.9375\ f_s \rightarrow 0.0625\ f_s$

$f_4 = 4\ f_1 = 1.2500\ f_s \rightarrow 0.2500\ f_s$

$f_5 = 5\ f_1 = 1.5625\ f_s \rightarrow 0.4375\ f_s$

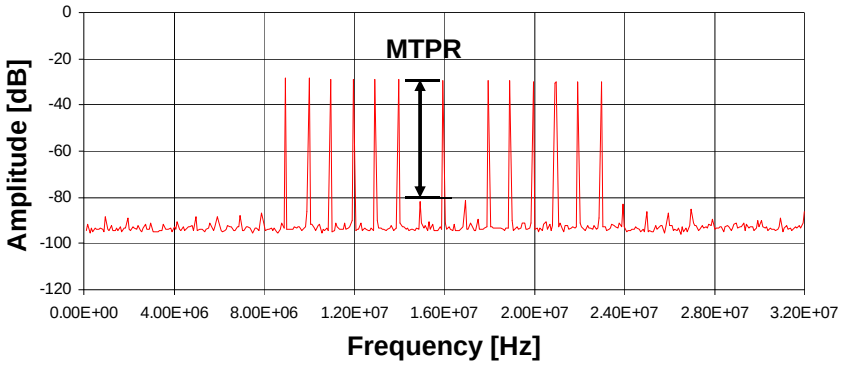


Intermodulation Distortion



- IMD is important in multi-channel communication systems
  - Third order products are generally difficult to filter out

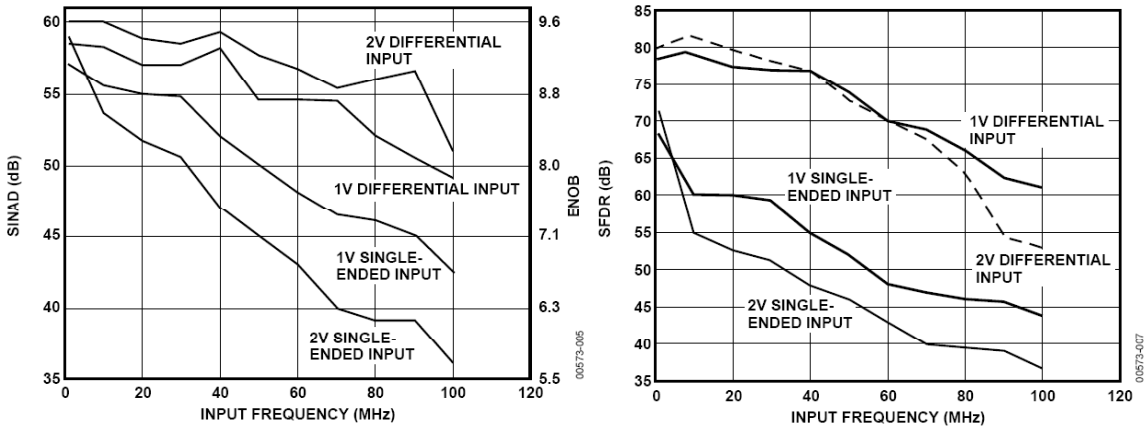
MTPR



- Useful metric in multi-tone transmission systems
  - E.g. OFDM

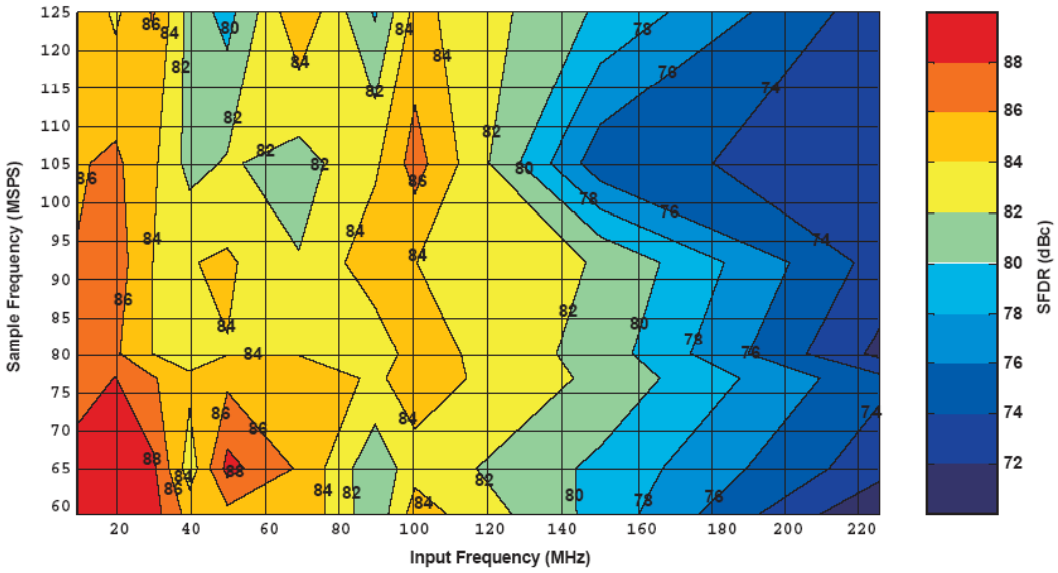
Frequency Dependence (1)

- All of the above discussed metrics generally depend on frequency
  - Sampling frequency and input frequency



[Analog Devices, AD9203 Datasheet]

Frequency Dependence (2)



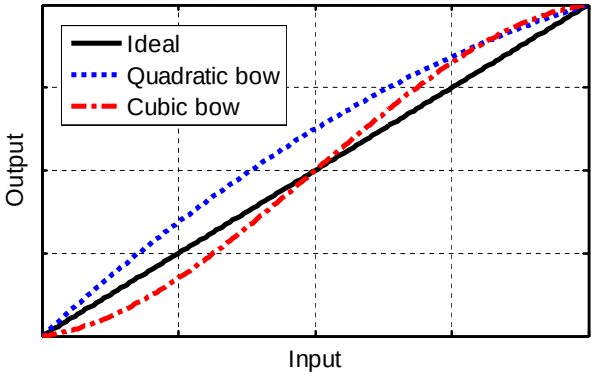
[Texas Instruments, ADS5541 Datasheet]

ERBW

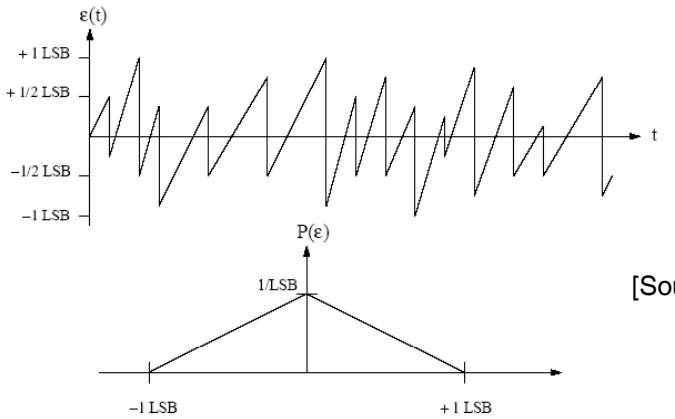
- Defined as the input frequency at which the SNDR of a converter has dropped by 3dB
  - Equivalent to a 0.5-bit loss in ENOB
- ERBW > $f_s/2$ is not uncommon, especially in converters designed for sub-sampling applications

Relationship Between INL and SFDR

- At low input frequencies, finite SFDR is mostly due to INL
- Quadratic/cubic bow gives rise to second/third order harmonic
- Rule of thumb: SFDR ≅ 20log(2^B/INL)
  - E.g. 1 LSB INL, 10 bits → SFDR ≅ 60dB



SNR Degradation due to DNL (1)



[Source: Ion Opris]

- For an ideal quantizer we assumed uniform quatization error over ±Δ/2
- Let's add uniform DNL over ± 0.5 LSB and repeat math...

SNR Degradation due to DNL (2)

- Integrate triangular pdf

$$\overline{e^2} = 2\int_0^{+\Delta}\left(1-\frac{e}{\Delta}\right)\frac{e^2}{\Delta}de = \frac{\Delta^2}{6} \qquad \Rightarrow SNR = 6.02\cdot B - 1.25 \ \ [dB]$$

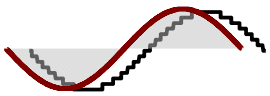
- Compare to ideal quantizer

$$\overline{e^2} = \int_{-\Delta/2}^{+\Delta/2}\frac{e^2}{\Delta}de = \frac{\Delta^2}{12} \qquad \Rightarrow SNR = 6.02\cdot B + 1.76 \ \ [dB]$$

3dB

- Bottom line: non-zero DNL across many codes can easily cost a few dB in SNR
  - "DNL noise"

Nyquist Rate DACs

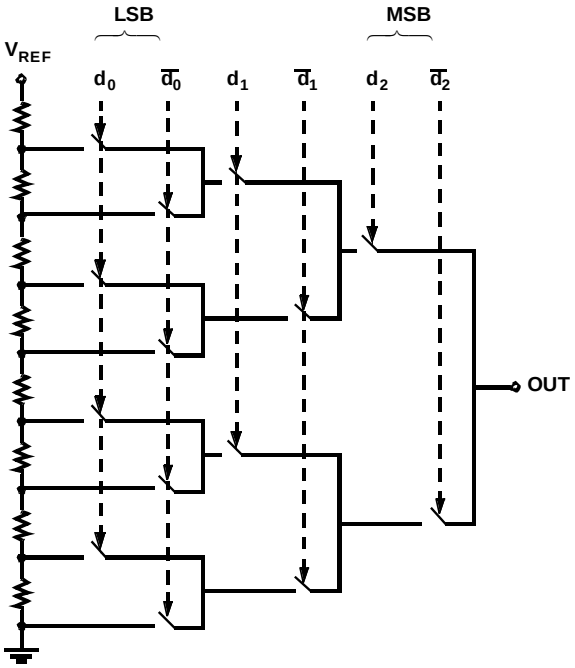


Katelijn Vleugels
Stanford University

Overview

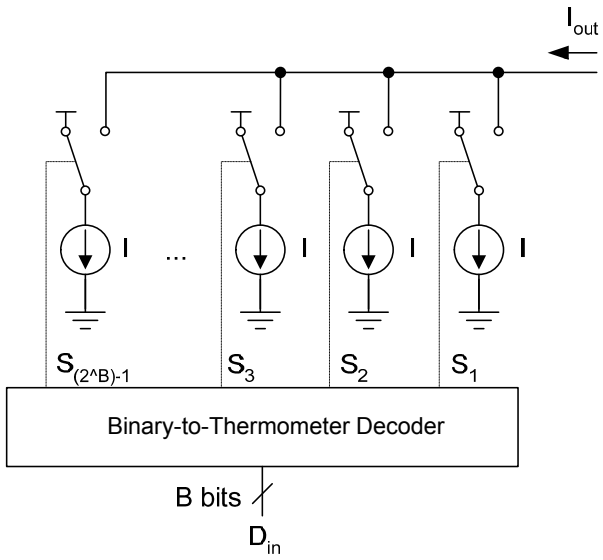
- D/A conversion is typically accomplished through the division or multiplication of a reference voltage, current or charge
- Architectures
  - Thermometer
  - Binary weighted
  - Segmented
- Static performance
  - Limited by component matching
- Dynamic performance
  - Limited e.g. by timing errors, "glitches"

Resistor String DAC



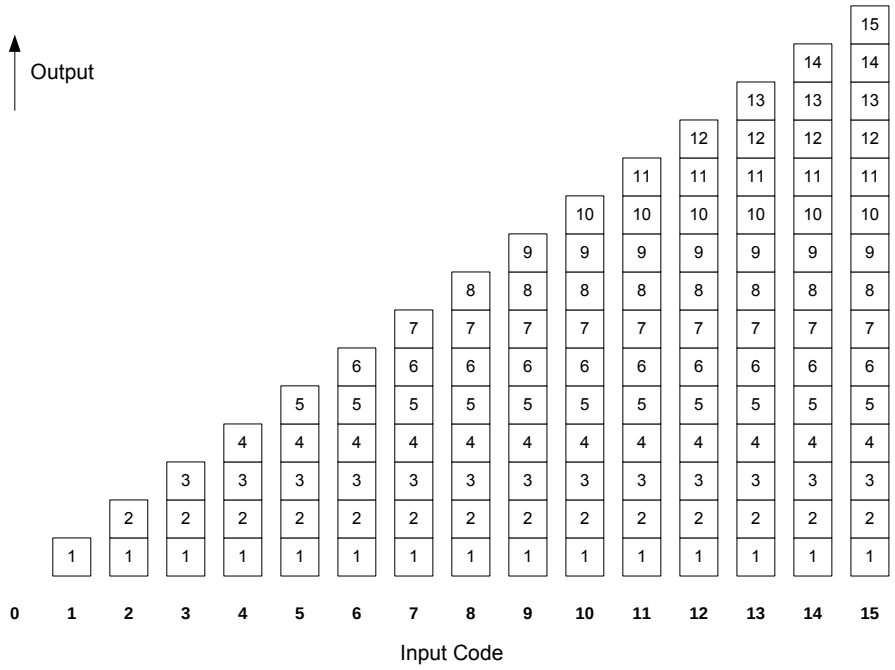
- Simple, inherently monotonic
- Small area up to ~8 bits
- See e.g. Pelgrom, JSSC 12/1990
- Unsuitable for high-resolution, high-speed designs

Thermometer DAC Using Switched Currents

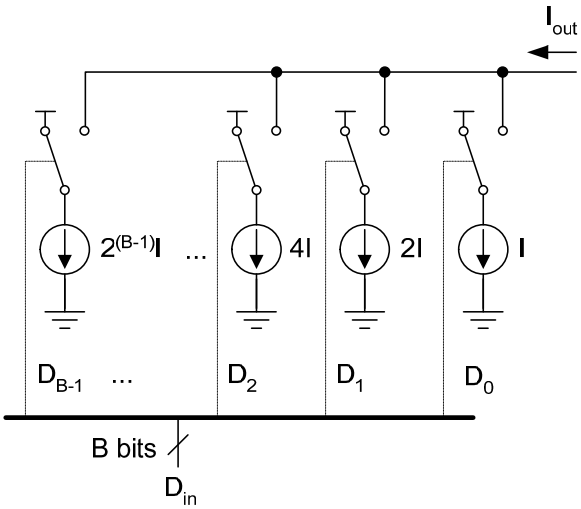


- Advantages
  - Monotonicity ensured by turning on equal-weighted current sources in succession
  - Fast: timing glitches proportional to step size do not degrade DNL
- Disadvantages
  - Impractical for large B (ie high resolution) due to large area taken up by current sources
  - Requires large decoder with 2^B-1 outputs

Thermometer DAC Principle



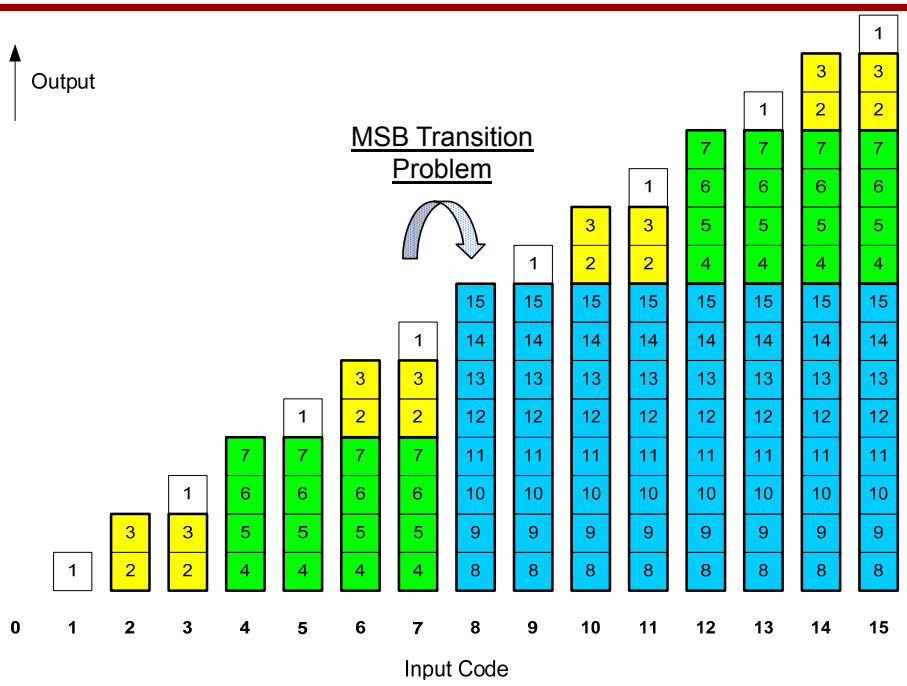
Binary Weighted DAC



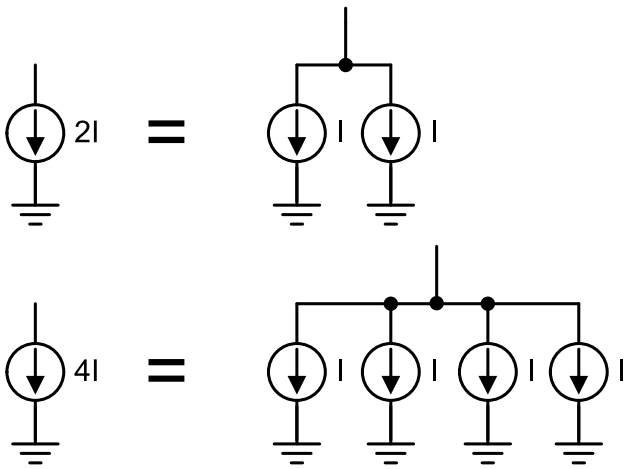
- Advantages
  - Area-efficient
  - No decoder needed
- Disadvantages
  - Monotonicity is not guaranteed: 2^k-1 source must match sum of 1-to-$2^{(k-1)}$ to within 1 LSB to make transition monotonic
  - Timing glitches introduce DNL

→ MSB transition is “worst case”

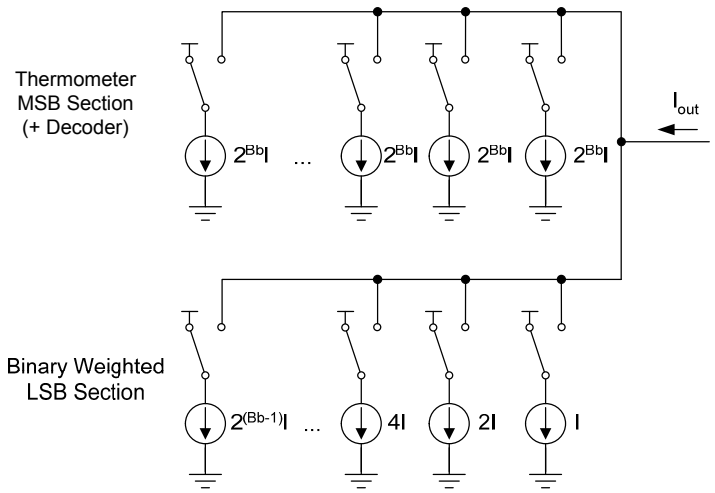
Binary Weighted DAC Principle



Implementation of Weighted Elements

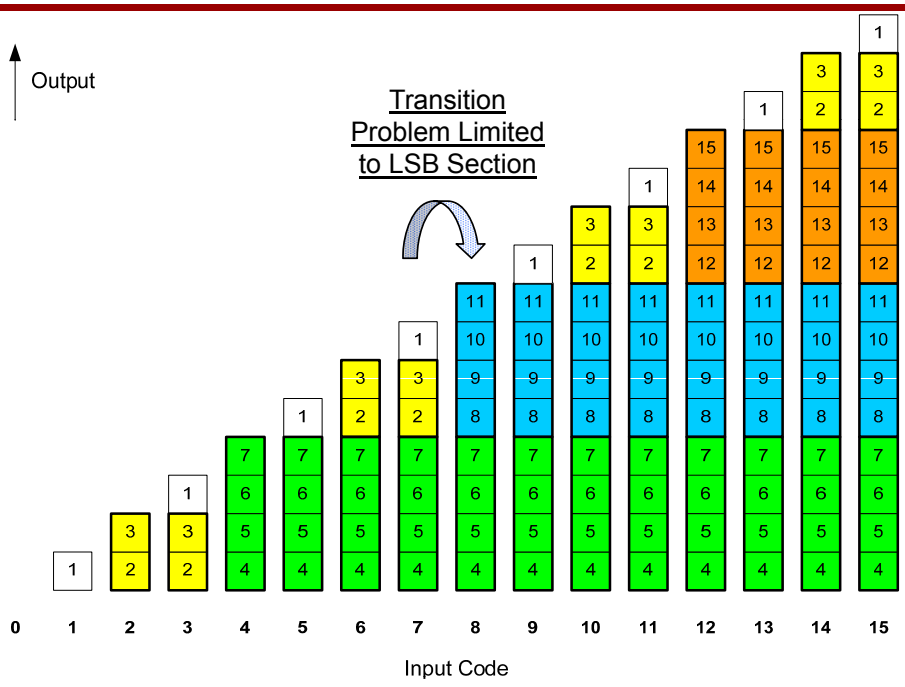


Segmented DAC



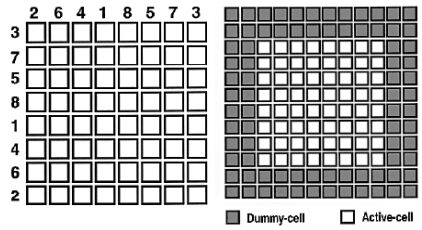
- Binary weighted section with B_b bits
- Thermometer section with $B_t = B-B_b$ bits
- Typically $B_t \sim 4\ldots8$
- Reasonably small decoder
- Easier to achieve monotonicity

Segmented DAC (2-2)

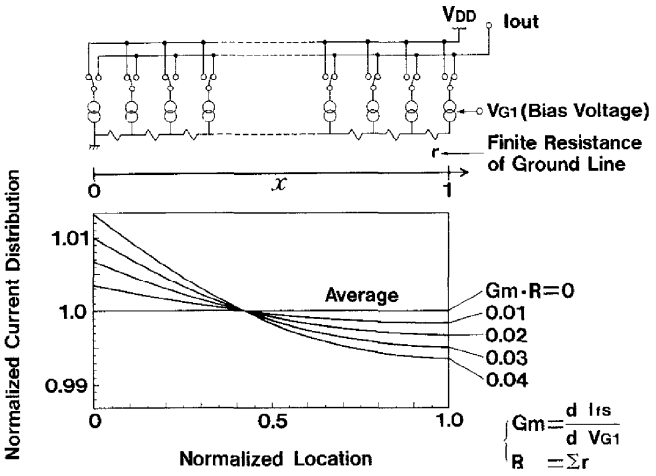


Static Errors (1)

- DNL and INL due to unit element mismatch
- Systematic Errors
  - Contact and wiring resistance (IR drop)
  - Edge effects in unit element arrays
  - Process gradients
  - Finite current source output resistance
- Systematic errors can be mitigated by proper layout and switching sequence design
  - See e.g. [Lin, JSSC 12/98], [Van der Plas, JSSC 12/99]



Mitigating IR Drop



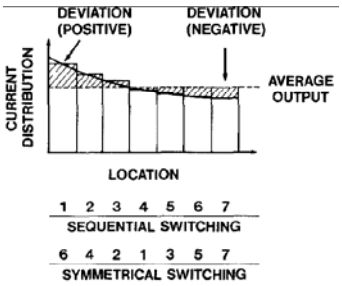
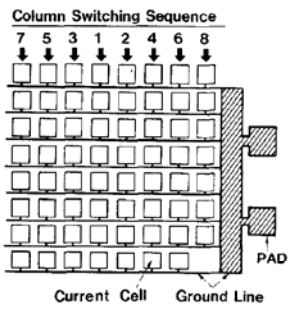


Fig. 9. Symmetrical switching.

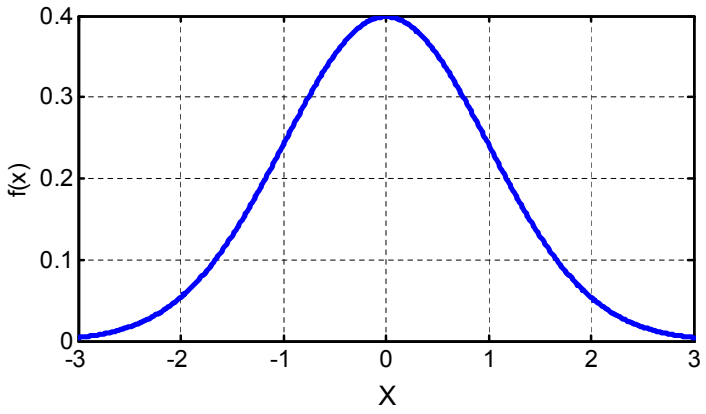


Static Errors (2)

- In the best possible scenario, we are then limited by “random” errors, which are due to material roughness, randomness in etching, etc.
- The distribution of random errors is usually well approximated by a Gaussian PDF (central limit theorem)
- References
  - C. Conroy et al., “Statistical Design Techniques for D/A Converters,” IEEE J. Solid-State Ckts., pp. 1118-28, Aug. 1989.
  - P. Crippa, et al., "A statistical methodology for the design of high-performance CMOS current-steering digital-to-analog converters," IEEE Trans. CAD of ICs and Syst. pp. 377-394, Apr. 2002.

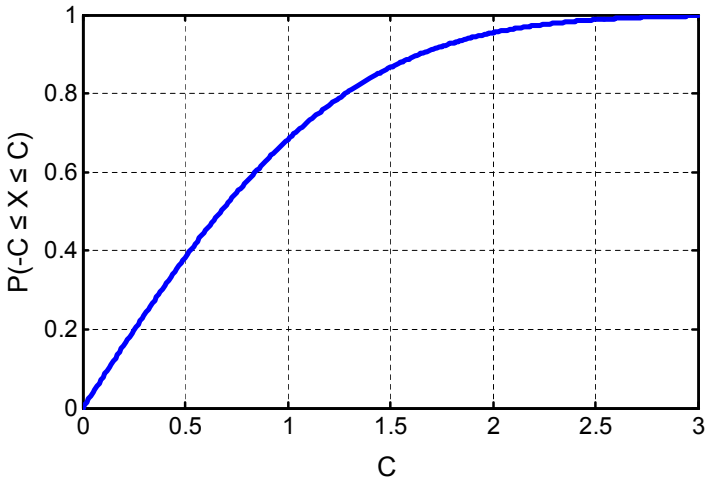
Gaussian Distribution

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad X = \frac{x-\mu}{\sigma}$$



Yield (1)

$$P(-C \le X \le +C) = \frac{1}{\sqrt{2\pi}} \int_{-C}^{+C} e^{-\frac{X^2}{2}} dX = \text{erf}\left(\frac{C}{\sqrt{2}}\right)$$



Yield (2)

C	P(-C ≤ X ≤ C) [%]	C	P(-C ≤ X ≤ C) [%]
0.2000	15.8519	2.2000	97.2193
0.4000	31.0843	2.4000	98.3605
0.6000	45.1494	2.6000	99.0678
0.8000	57.6289	2.8000	99.4890
1.0000	68.2689	3.0000	99.7300
1.2000	76.9861	3.2000	99.8626
1.4000	83.8487	3.4000	99.9326
1.6000	89.0401	3.6000	99.9682
1.8000	92.8139	3.8000	99.9855
2.0000	95.4500	4.0000	99.9937

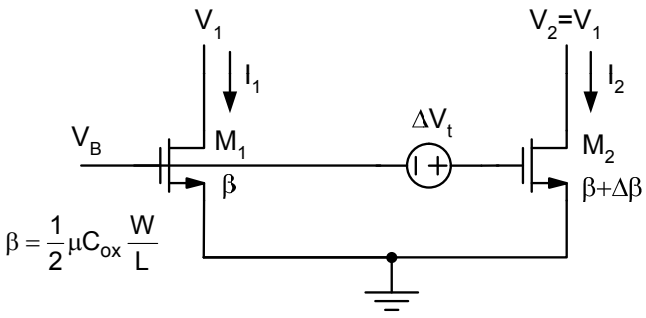
Example

- Measurements show that the current in a production lot of current sources follows a Gaussian distribution with σ = 0.1 mA and μ = 10 mA
  - What fraction of current sources is within ±3% (or ±1%) of the mean?
- Relative matching ("coefficient of variation")

$$\sigma_u = \frac{\sigma}{\mu} = \text{stdev}\left(\frac{\Delta I}{I}\right) = \frac{0.1\text{mA}}{10\text{mA}} = 1\%$$

- Fraction of current sources within 3%
  - C = 3 → 99.73%
- Fraction of current sources within 1%
  - C = 1 → 68.27%

Mismatch in MOS Current Sources



$$\Delta I = I_1 - I_2 \cong -g_m \Delta V_t + I_1 \frac{\Delta\beta}{\beta}$$

$$\frac{\Delta I}{I_1} \cong -\frac{g_m}{I_1}\Delta V_t + \frac{\Delta\beta}{\beta}$$

$$\sigma_{\Delta V_t} = \frac{A_{V_t}}{\sqrt{WL}} \qquad \sigma_{\frac{\Delta\beta}{\beta}} = \frac{A_\beta}{\sqrt{WL}}$$

- Example
  - W=500μm, L=0.2μm, g_m/I_D=10S/A, A_{Vt}=5mV-μm, A_β=1%-μm

$$\sigma_{\frac{\Delta I}{I_1}} = \sqrt{\left(10\frac{S}{A}\cdot\frac{5\text{mV}}{10}\right)^2 + \left(\frac{1\%}{10}\right)^2} = \sqrt{(0.5\%)^2 + (0.1\%)^2} = 0.51\%$$

DNL of Thermometer DAC

$$DNL(k) = \frac{Step(k) - Step_{avg}}{Step_{avg}} = \frac{I_k - \frac{1}{N}\sum_{j=1}^{N} I_j}{\frac{1}{N}\sum_{j=1}^{N} I_j} \cong \frac{I_k - I}{I} = \frac{\Delta I}{I}$$

$$stdev\left(DNL(k)\right) = stdev\left(\frac{\Delta I}{I}\right) = \sigma_u$$

- Standard deviation of DNL for each code is simply equal to relative matching (σ_u) of unit elements
- Example
  - Say we have unit elements with σ_u = 1% and want 99.73% of all converters to meet the spec
  - Which DNL specification value should go into the datasheet?

DNL Yield Example (1)

- First cut solution
  - For 99.73% yield, need C = 3
  - $\sigma_{DNL} = \sigma_u$ = 1%
  - 3 σ_{DNL} = 3%
  - DNL specification for a yield of 99.73% is ±0.03 LSB
    - Independent of target resolution (?)
- Not quite right
  - Must keep in mind that a converter will meet specs only if all codes meet DNL spec, i.e. DNL(k) < DNL_{spec} for all k
  - A converter with more codes is less likely to have all codes meet the specification
  - Let's see if this is significant

DNL Yield Example (2)

- Let's say there are N codes, and assume that all DNL(k) values are independent, then
  - P(all codes meet spec) = P(single code meets spec)^N
  - P(all codes meet spec)$^{1/N}$ = P(single code meets spec)
- Lets look at two examples N=63 (6 bits) and N=4095 (12 bits)
  - $0.9973^{1/63}$ = 0.99995708…
  - $0.9973^{1/4095}$ = 0. 99999929929…
- Can calculate modified confidence intervals using Matlab
  - For N=63, C = sqrt(2)*erfinv($0.9973^{1/63}$) = 4.09
  - For N=4095, C = sqrt(2)*erfinv($0.9973^{1/4095}$) = 4.97
- Refined result for 99.97% yield
  - N=63: DNL spec should be ±0.0409 LSB
  - N=4095: DNL spec should be ±0.0497 LSB

DNL Yield Example (3)

- Getting a more accurate yield estimate for the preceding example wasn't all that hard
  - Unfortunately things won't always be that simple
    - E.g. in a segmented DAC, DNL(k) are no longer independent
- The "typical" DAC designer tends to rely on simulations rather than trying to formulate "exact" yield equations
  - Get rough estimate using simple (often optimistic) expressions
  - Run "Monte Carlo" simulations in Matlab to find actual yield or to center specs
  - Still important to have a qualitative feel for what may cause discrepancies

INL of Thermometer DAC (1)

$$INL(k) = \frac{I_{out}(k) - I_{out,uniform}(k)}{Step_{avg}}$$

$$= \frac{\sum_{j=1}^{k} I_j - \frac{k}{N}\sum_{j=1}^{N} I_j}{\frac{1}{N}\sum_{j=1}^{N} I_j} = \frac{N\sum_{j=1}^{k} I_j}{\sum_{j=1}^{k} I_j + \sum_{j=k+1}^{N} I_j} - k$$

$$= \frac{A \cdot N}{A + B} - k$$

$$\mathrm{var}\left(INL(k)\right) = \mathrm{var}\left(\frac{A \cdot N}{A + B} - k\right) = N^2\,\mathrm{var}\left(\frac{A}{A + B}\right) = N^2\,\mathrm{var}\left(\frac{X}{Y}\right)$$

INL of Thermometer DAC (2)

- For a quotient of random variables

$$\mathrm{var}\left(\frac{X}{Y}\right) \cong \left(\frac{\mu_X}{\mu_Y}\right)^2 \left(\frac{\sigma_X^2}{\mu_X^2} + \frac{\sigma_Y^2}{\mu_Y^2} - 2\frac{\mathrm{cov}(X,Y)}{\mu_X \mu_Y}\right)$$

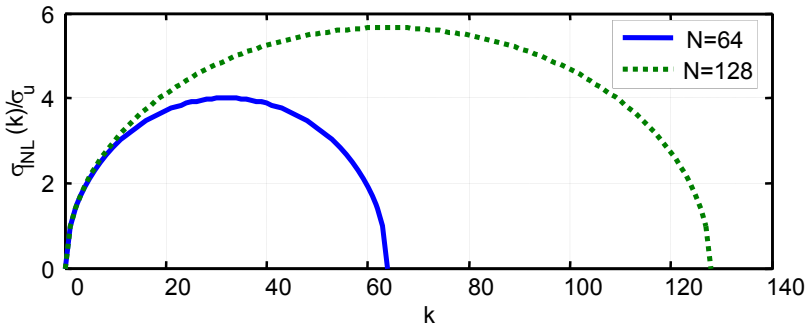
[Dennis E. Blumenfeld, *Operations Research Calculations Handbook*, Online: http://www.engnetbase.com/ejournals/books/book_summary/toc.asp?id=701]

- After identifying the means (μ), variances (σ^2) and covariance (*cov*) needed in the above approximation, it follows that

$$\mathrm{var}\left(INL(k)\right) \cong k\left(1 - \frac{k}{N}\right)\sigma_u^2$$

$$\sigma_{INL}(k) \cong \sigma_u \sqrt{k\left(1 - \frac{k}{N}\right)}$$

INL of Thermometer DAC (3)



- Standard deviation of INL is maximum at mid-scale (k=N/2)

$$\sigma_{INL} \cong \sigma_u \sqrt{\frac{N}{2}\left(1-\frac{N/2}{N}\right)} = \frac{1}{2}\sigma_u\sqrt{N} \cong \frac{1}{2}\sigma_u\sqrt{2^B}$$

- For a more elaborate derivation of this result see [Kuboki et al., IEEE Trans. Circuits & Systems, 6/1982]

Achievable Resolution

$$B \cong \log_2\left(4\left[\frac{\sigma_{INL}}{\sigma_u}\right]^2\right) = 2 + 2\log_2\left(\frac{\sigma_{INL}}{\sigma_u}\right)$$

- Example: σ_{INL}= 0.1 LSB (at mid-scale code)

σ_u	B
1%	8.6
0.5%	10.6
0.2%	13.3
0.1%	15.3

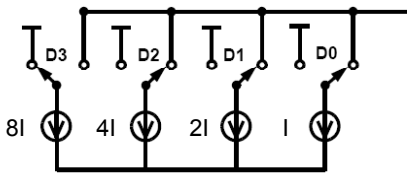
INL Yield

- Again, we should ask how many DACs will meet the spec for a given σ_{INL} (worst code)
  - It turns out that this is a very difficult math problem
- Two solutions
  - Do the math
    - G. I. Radulov et al., "Brownian-Bridge-Based Statistical Analysis of the DAC INL Caused by Current Mismatch," IEEE TCAS II, pp. 146-150, Feb. 2007.
  - Yield simulations
- Good rule of thumb
  - For high target yield (>95%), the probability of "all codes meet INL spec" is very close to "worst code meets INL spec"

DNL/INL of Binary Weighted DAC

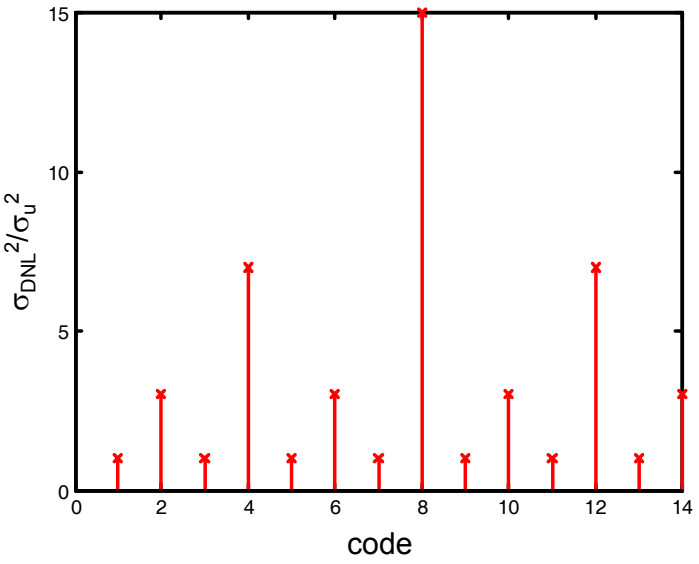
- INL same as for thermometer DAC
  - Why?
- DNL is not same for all codes, but depends on transition
- Consider worst case: 0111 … → 1000 …
  - Turning on MSB and turning off all LSBs

$$\sigma_{DNL}^2 = \underbrace{\left(2^{B-1}-1\right)\sigma_u^2}_{0111...} + \underbrace{\left(2^{B-1}\right)\sigma_u^2}_{1000...} = \left(2^B-1\right)\sigma_u^2$$

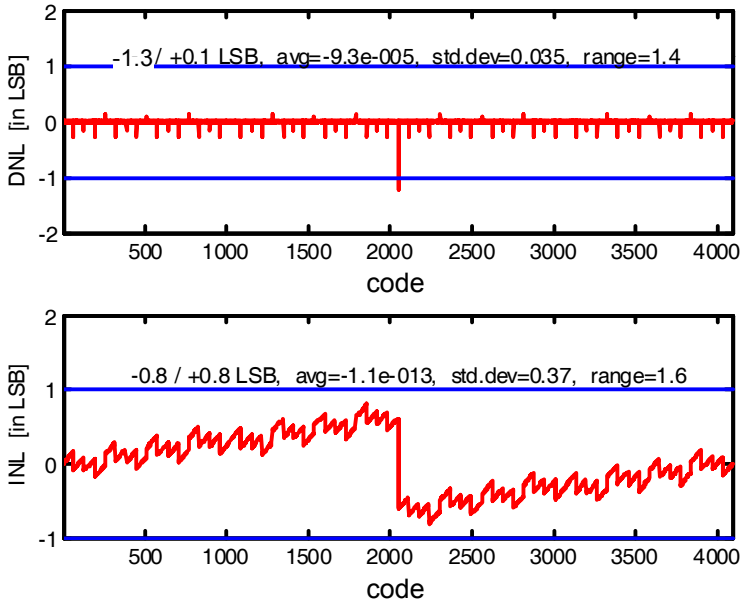


- Example
  - B = 12, σ_u = 1% → σ_{DNL} = 0.64 LSB
  - Much worse than thermometer DAC

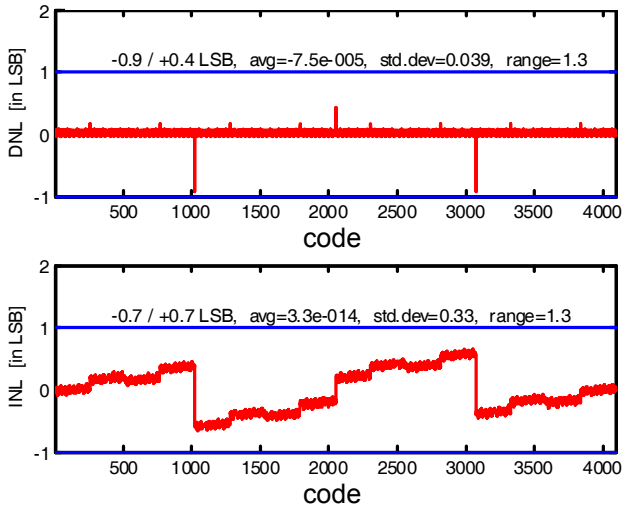
σ_{DNL} (4-bit Example)



Simulation Example

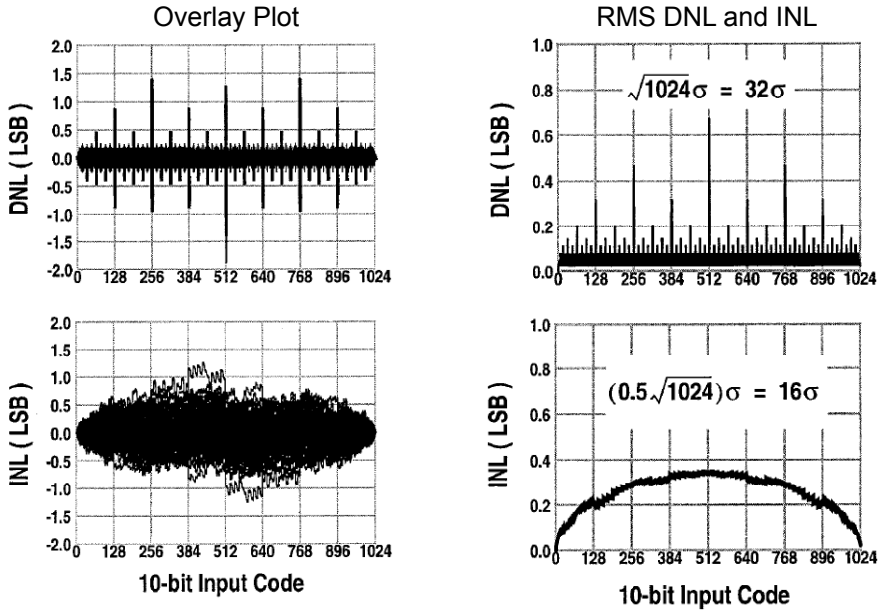


Another Random Run



- Peak DNL not at mid-scale!
  - Important to realize that this is just one single statistical outcome…

Multiple Simulation Runs (100)

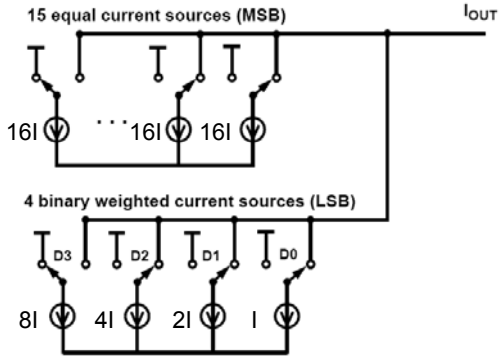


[Lin & Bult, JSSC 12/1998]

DNL/INL of Segmented DAC

- INL
  - Same as in thermometer DAC
- DNL
  - Worst case occurs when LSB DAC turns off and one more MSB DAC element turns on
  - Essentially same DNL as a binary weighted DAC with B_b+1 bits

Example: $B=B_b+B_t=4+4=8$



Comparison

	Thermometer	Segmented	Binary Weighted
σ_{INL} (worst case)	$\cong \frac{1}{2}\sigma_u\sqrt{2^B}$		
σ_{DNL} (worst case)	$\cong \sigma_u$	$\cong \sigma_u\sqrt{2^{B_b+1}-1}$	$\cong \sigma_u\sqrt{2^B-1}$
Number of Switched Elements	2^B-1	$B_b+2^{B_t}-1$	B

Example (B=12, σ_u=1%)

DAC Architecture	σ_{INL} (worst)	σ_{DNL} (worst)	Number of Switched Elements
Thermometer	0.32	0.01	4095
Binary Weighted	0.32	0.64	12
Segmented (B_b=7, B_t=5)	0.32	0.16	38

Dynamic DAC Errors (1)

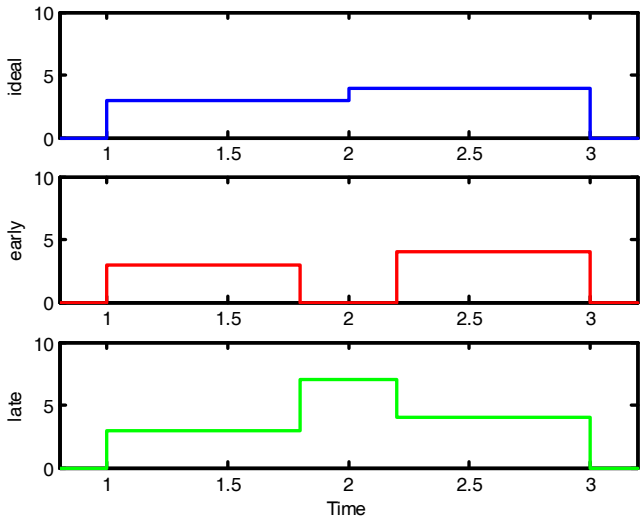
- Finite settling time and slewing
  - Finite RC time constant
  - Signal dependent slewing
- Feedthrough
  - Coupling from switch signals to DAC output
  - Clock feedthrough
- Glitches due to timing errors
  - Current sources won't switch simultaneously
- Dynamic DAC errors are generally hard to model!

Dynamic DAC Errors (2)

- References
  - Gustavsson, Chapter 12
  - M. Albiol, J.L. Gonzalez, E. Alarcon, "Mismatch and dynamic modeling of current sources in current-steering CMOS D/A converters," IEEE TCAS I, pp. 159-169, Jan. 2004
  - Doris, van Roermund, Leenaerts, Wide-Bandwidth High Dynamic Range D/A Converters, Springer 2006.
  - T. Chen and G.G.E. Gielen, "The analysis and improvement of a current-steering DAC's dynamic SFDR," IEEE Trans. Ckts. Syst. I, pp. 3-15, Jan. 2006.

Glitch Impulse (1)

- DAC output waveform depends on timing
  - Consider binary weighted DAC transition 0111… → 1000…



Ideal

LSBs early, MSB late

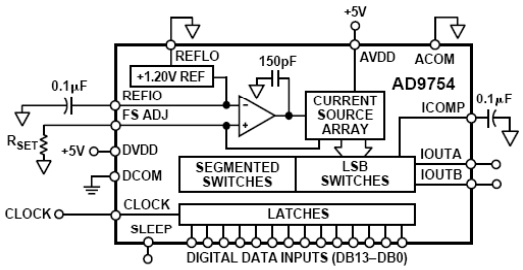
LSBs late, MSB early

Glitch Impulse (2)

- Worst case glitch impulse (area): $\propto \Delta t\, 2^{B-1}$
- LSB area: $\propto T$
- Need $\Delta t\, 2^{B-1} << T$ which implies $\Delta t << T/2^{B-1}$

f_s [MHz]	B	Δt [ps]
1	12	<< 488
20	16	<< 1.5
1000	10	<< 2

Commercial Example



AD9754

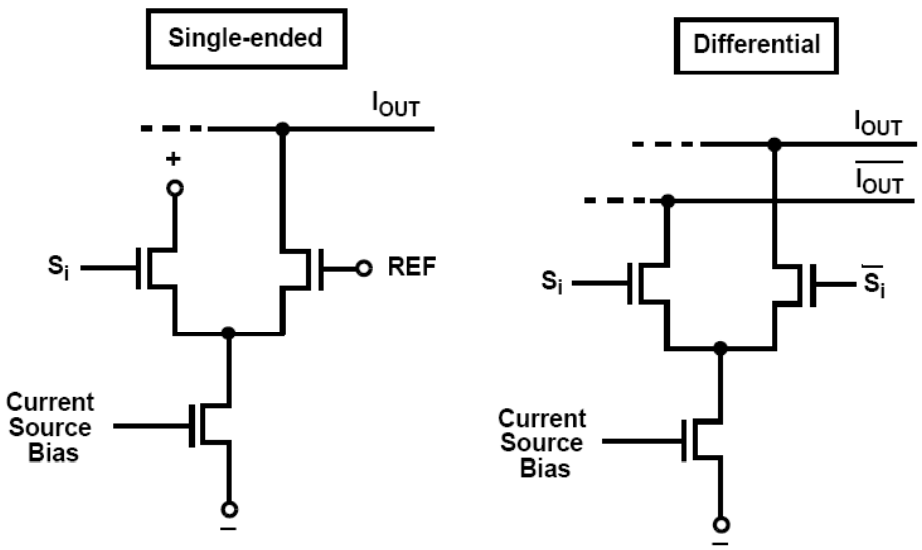
DYNAMIC SPECIFICATIONS (T_{MIN} to T_{MAX}, AVDD = +5 V, DVDD = +5 V, I_{OUTFS} = 20 mA, Differential Transformer Coupled Output, 50 Ω Doubly Terminated, unless otherwise noted)

Parameter	Min	Typ	Max	Units
DYNAMIC PERFORMANCE				
Maximum Output Update Rate (f_{CLOCK})	125			MSPS
Output Settling Time (t_{ST}) (to 0.1%)[1]		35		ns
Output Propagation Delay (t_{PD})		1		ns
→ Glitch Impulse		5		pV-s
Output Rise Time (10% to 90%)[1]		2.5		ns
Output Fall Time (10% to 90%)[1]		2.5		ns
Output Noise (I_{OUTFS} = 20 mA)		50		pA/√Hz
Output Noise (I_{OUTFS} = 2 mA)		30		pA/√Hz

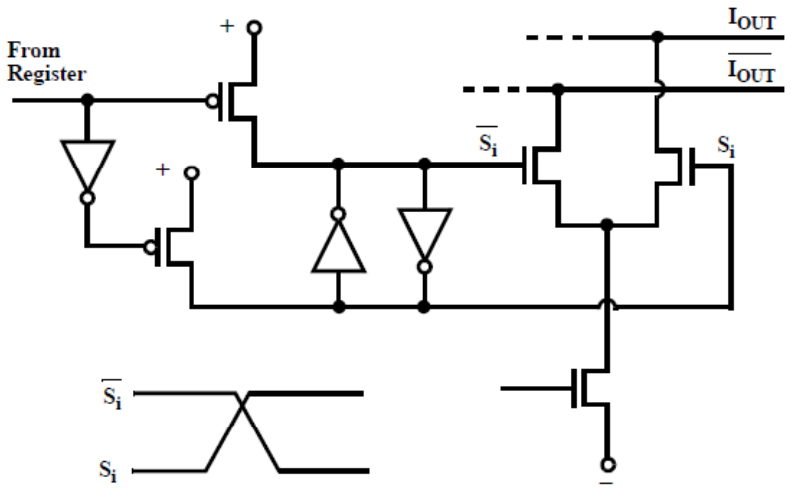
How to minimize dynamic DAC errors?

- Differential pair switch
- Retiming
  - Latches in (or close to) each current cell
  - Latch controlled by global clock to ensure that current cells switch simultaneously (independent of decoder delays)
- Make before break
  - Ensure uninterrupted current flow, so that tail current source remains active
- Low swing driver
  - Drive differential pair with low swing to minimize coupling from control signals to output
- Cascoded tail current source for high output impedance
  - Ensures that overall impedance at output nodes is code independent (necessary for good INL)

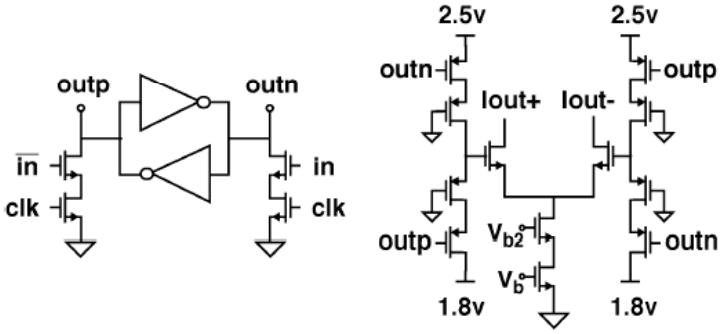
Basic Differential Pair Switch



Make-Before-Break Driver



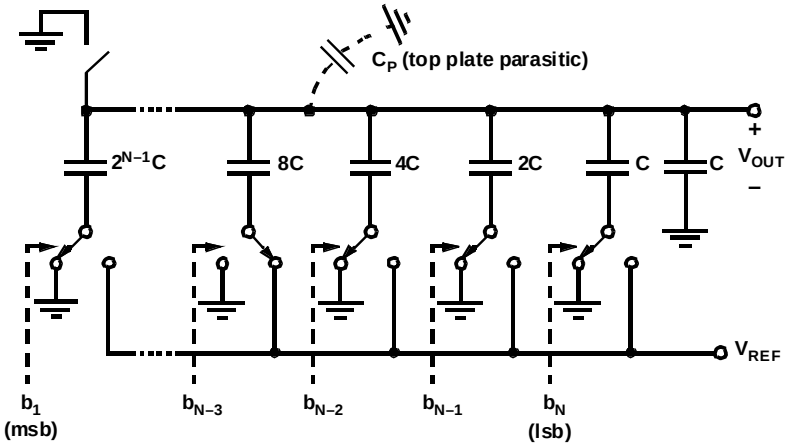
Example Current Cell Implementation



[Barkin & Wooley, JSSC 4/2004]

- Differential pair switch
- Make-before-break driver
- Low swing driver
- Cascoded tail current source

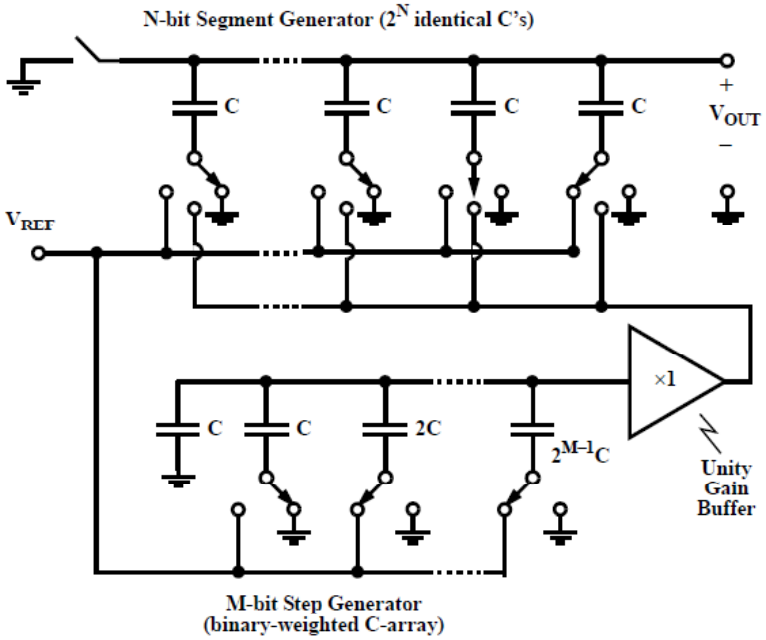
Binary Weighted Charge Redistribution DAC



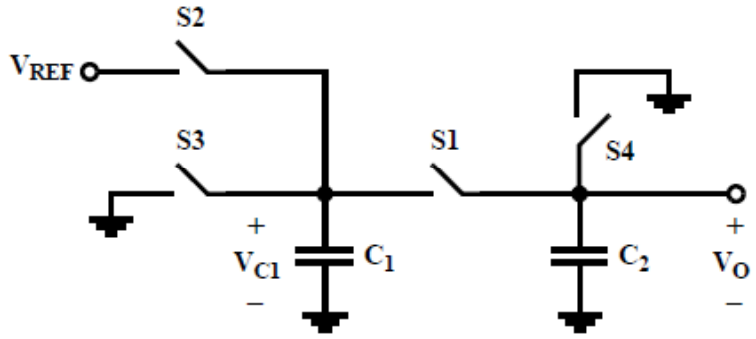
$$V_{out} = \frac{2^B C}{2^B C + C_p} \cdot V_{ref} \sum_{i=1}^{B} \frac{b_i}{2^i}$$

- Can redistribute charge onto OTA + feedback capacitor to mitigate gain error due to C_p

Segmented Charge Redistribution DAC



Serial Charge Redistribution DAC



Implementation Example Segmented DAC

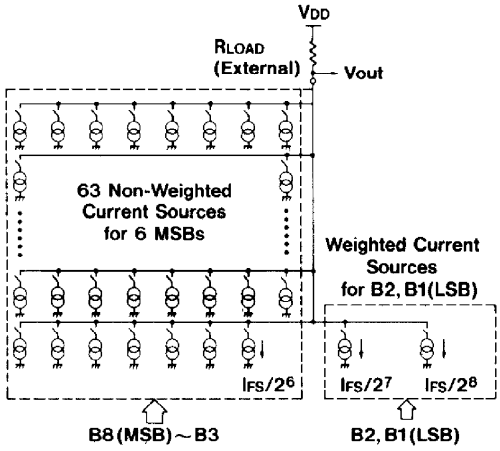


Fig. 1. Basic architecture of the DAC.

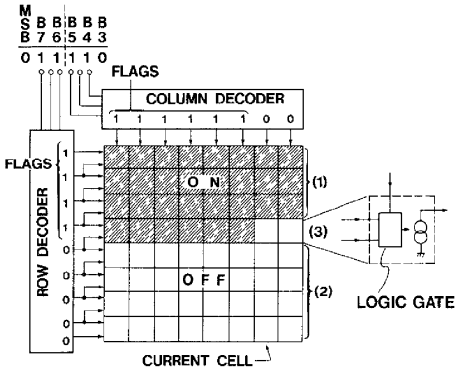
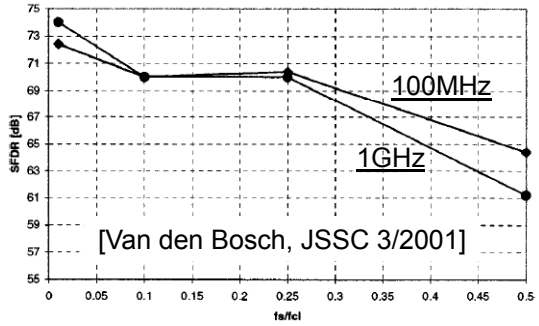


Fig. 2. Two-step decoding.

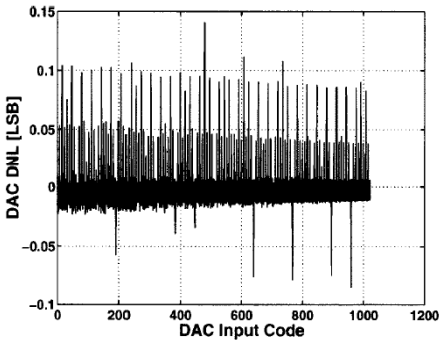
[T. Miki, Y. Nakamura, M. Nakaya, S. Asai, Y. Akasaka, and Y. Horiba, "An 80-MHz 8-bit CMOS D/A Converter," IEEE J. of Solid-State Circuits, pp. 983-988, Dec. 1986.]

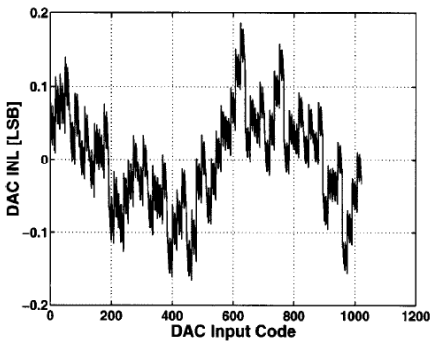
High Performance DAC Examples (1)



[Van den Bosch, JSSC 3/2001]

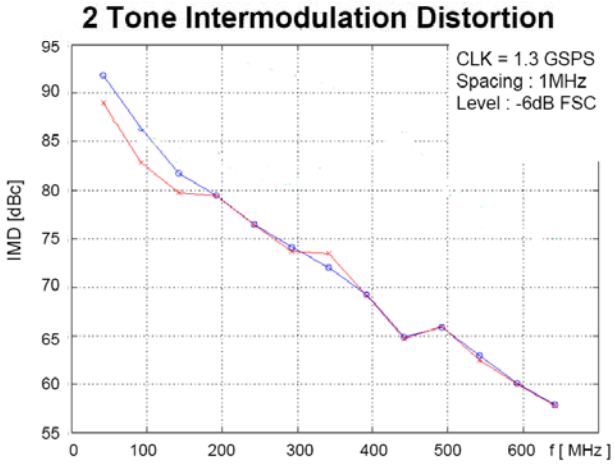
Resolution	10 bit
Update rate	Up to 1 GS/s
INL	< 0.2LSB
DNL	<0.15 LSB
SFDR (490MHz@1GS/s)	61.2 dB
Analog/digital voltage supply	3V / 1.9 V
Power consumption	110mW (490Mhz@1GHz)
Active area	0.35 mm^2
Process	0.35 µm CMOS





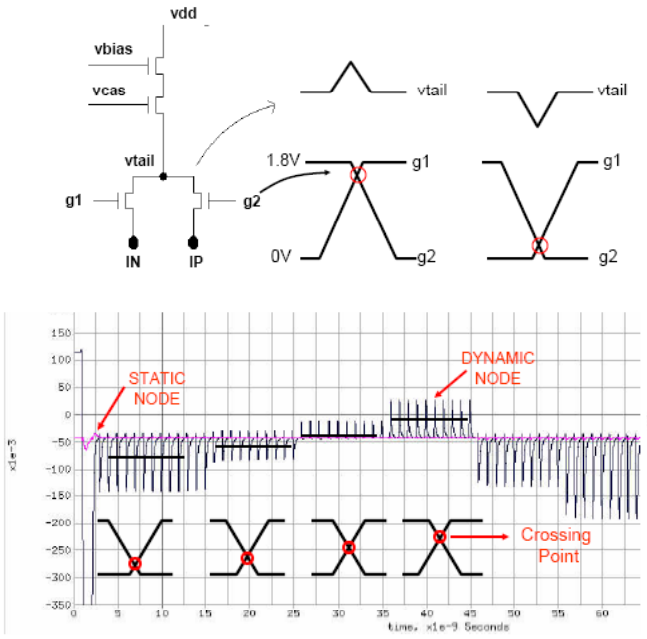
High Performance DAC Examples (2)

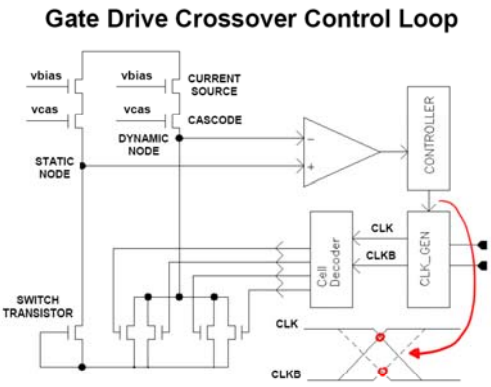
Max Sample Frequency	1.4	GSPS
Resolution	14	Bit
DNL	+/- 0.8	LSB
INL	+/- 2.1	LSB
SFDR @ 1.0 GSPS	> 60	dB
IMD @ 1.0 GSPS	> 64	dBc
NSD @ f_{out} = 400MHz	-155	dBm/Hz
Power (Core) @1.4GSPS	200	mW
Power(Total) @ 1.4GSPS	400	mW
Area (Core)	0.8	mm^2
Area (Chip)	6.25	mm^2



[Schafferer, ISSCC 2004]

High Performance DAC Examples (3)

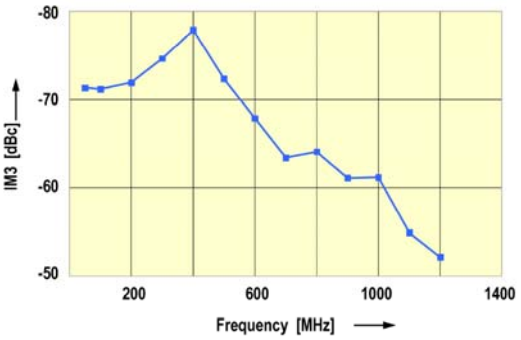


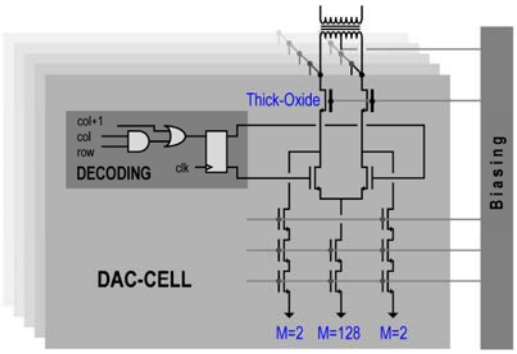


[Schafferer, ISSCC 2004]

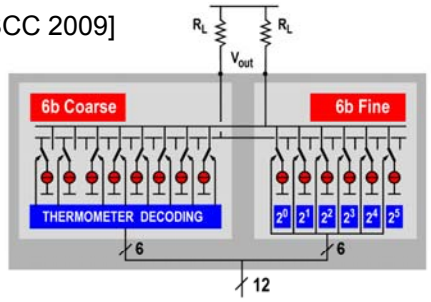
High Performance DAC Examples (4)

Technology	65nm CMOS
Number of Bits	12
Conversion speed	2.9 GS/s
INL	0.5 lsb
DNL	0.3 lsb
R_{load}	50 Ω
$I_{tail,total}$	50 mA
Voltage Swing	2.5 V_{ppd}
Power Dissipation	188 mW
Area	0.31 mm^2

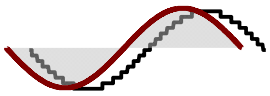




[Lin, ISSCC 2009]

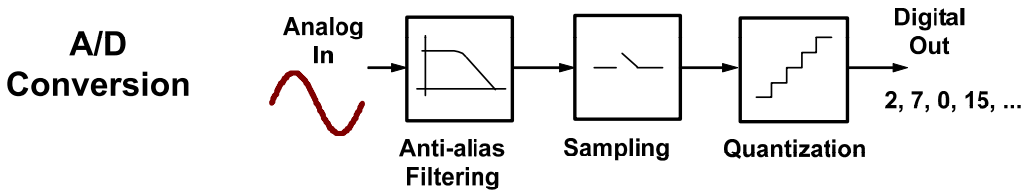


Sample & Hold Circuits



Katelijn Vleugels
Stanford University

Recap

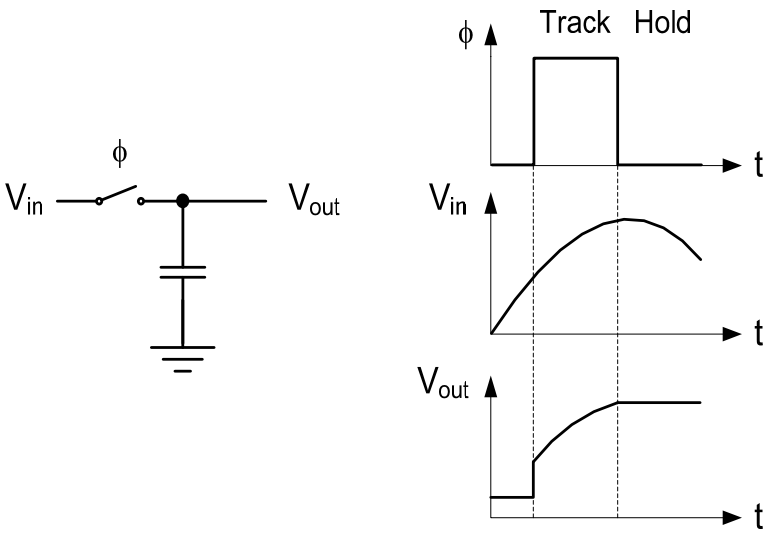


- How to build circuits that "sample"?
- Ideal Dirac sampling is impractical
  - Need a switch that opens, closes and acquires signal within an infinitely small time
- Practical solution
  - "Track and hold"

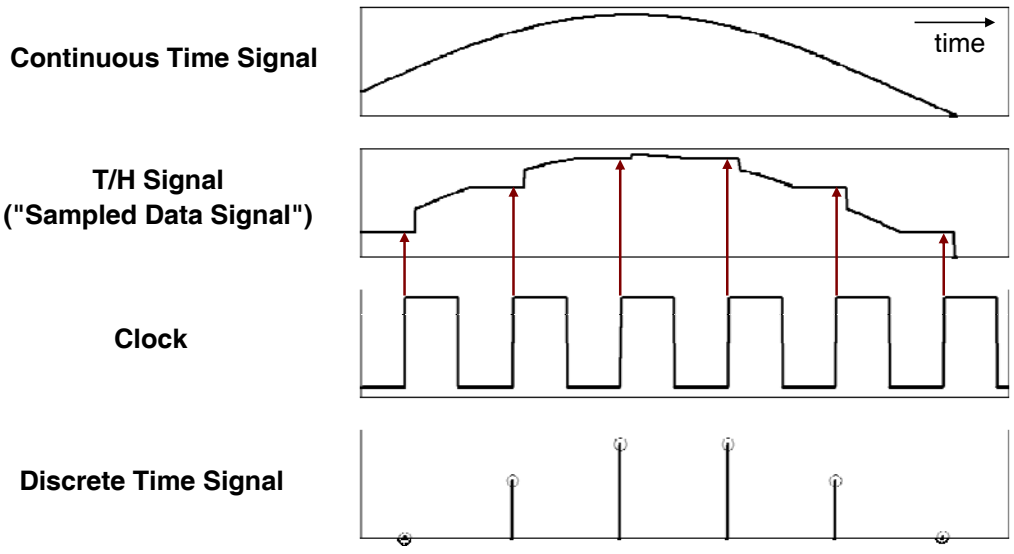
Outline

- Elementary track-and-hold circuit and its nonidealities
- First order improvements to elementary track-and-hold
- Advanced techniques
  - Clock bootstrapping
  - Bottom plate sampling
- Settling and noise analysis in charge-redistribution track-and-hold circuit
- Noise simulation example

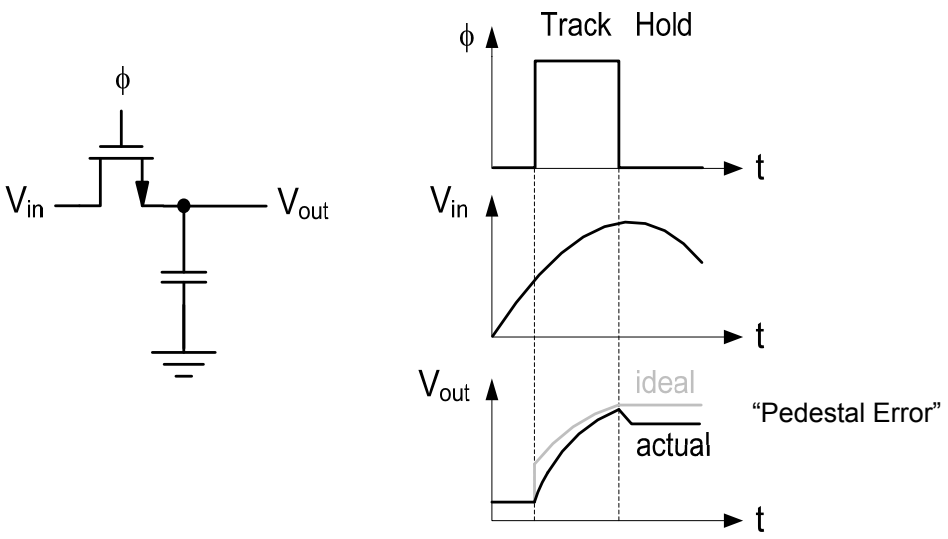
Ideal Track-and-Hold Circuit



Signal Nomenclature



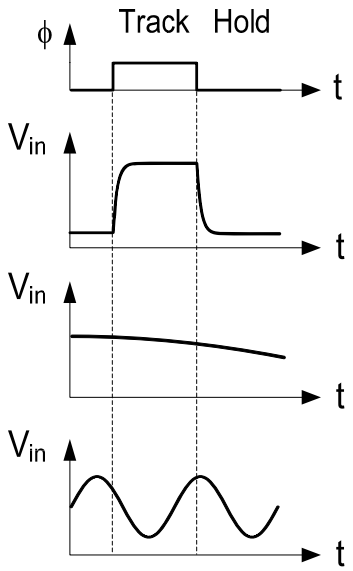
Circuit with MOS Switch



Nonidealities

- Non-zero acquisition time / finite bandwidth
- Thermal noise
- Aperture uncertainty / clock jitter
- Signal dependent sampling instant
- Tracking nonlinearity
- Hold mode feedthrough
- Hold mode leakage
- Charge injection and clock feedthrough

Non-zero Acquisition Time

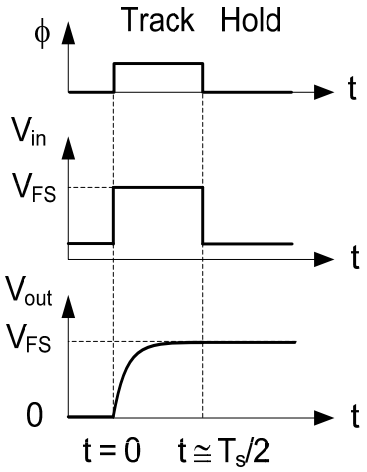


- Consider various input signal scenarios

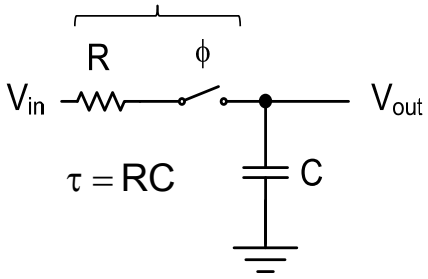
1. Input is a sampled data signal, i.e. the output of another switched capacitor stage
2. Input is a slowly varying continuous time signal, e.g. the input of an oversampling ADC
3. Input is a rapidly varying continuous time signal, e.g. the input of a Nyquist or sub-sampling ADC

Non-zero Acquisition Time – Case 1

- For simplicity, neglect finite rise time of the input signal
- Consider worst case – the output is required to settle from 0 to the full-scale voltage of the system (V_{FS})



First order MOS switch model



$$\Rightarrow V_{out}(t) = V_{FS}\left(1 - e^{-t/\tau}\right)$$

Non-zero Acquisition Time – Case 1

$$V_{out,err}\left(\frac{T_s}{2}\right) = -V_{FS}e^{-\frac{T_s}{2}/\tau} = -V_{FS}e^{-N}$$

- Typically think about settling error in terms of the number of settling time constants (N) required for ½ LSB settling in a B-bit system

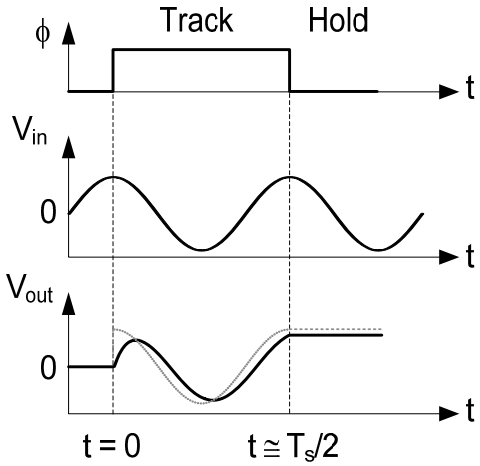
$$\left|-V_{FS}e^{-N}\right| \le \frac{1}{2}\frac{V_{FS}}{2^B}$$

$$\boxed{N = \frac{T_s/2}{\tau} \ge \ln\left(2 \cdot 2^B\right)}$$

B	N
6	>4.9
10	>7.6
14	>10.4
18	>13.2

Non-zero Acquisition Time – Case 2 & 3

- Consider a sinusoidal input around “0”, and “0” also as the initial condition for V_{out} (for notational simplicity)



$$V_{in}(t) = A\cos(\omega t + \phi)$$

$$V_{out}(t) = \underbrace{-\frac{A\cos(\phi-\theta)}{\sqrt{1+\omega^2\tau^2}}e^{-\frac{t}{\tau}}}_{\text{initial transient}} + \underbrace{\frac{A\cos(\omega t+\phi-\theta)}{\sqrt{1+\omega^2\tau^2}}}_{\text{steady-state response}}$$

$$\theta = \text{atan}(\omega\tau)$$

Non-zero Acquisition Time – Case 2 & 3

- At $t=T_s/2$, the error in the held signal consists of two parts
- Residual error due to initial exponentially decaying initial transient term
  - In order to minimize this error, we need to chose N appropriately, as calculated for the step input scenario
- Error due to magnitude attenuation and phase shift in the steady state term
  - This error depends only on the RC time constant and the input frequency; it cannot be reduced by extending the length of the track phase
  - How significant is the error due to the steady-state term?

Non-zero Acquisition Time – Case 2 & 3

- As an example, let's compute the percent amplitude error for the N values derived previously (½ LSB, B-bit settling to a step)

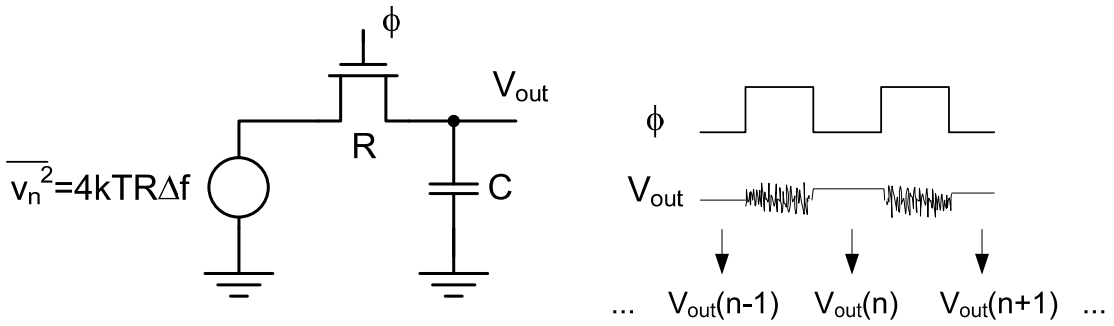
$$A_{err} = \left| \frac{\frac{A}{\sqrt{1+(\omega\tau)^2}} - A}{A} \right| = 1 - \frac{1}{\sqrt{1+(\omega\tau)^2}} = 1 - \frac{1}{\sqrt{1+\left(2\pi f_{in}\frac{T_s/2}{N}\right)^2}} = 1 - \frac{1}{\sqrt{1+\left(\frac{\pi}{N}\frac{f_{in}}{f_s}\right)^2}}$$

B	N	A_{err} ($f_{in} = f_s/20$)	A_{err} ($f_{in} = f_s/2$)
6	4.9	0.052%	4.9%
10	7.6	0.021%	2.1%
14	10.4	0.011%	1.1%
18	13.2	0.007%	0.7%

Summary – Non-zero Acquisition Time

- Precise settling to an input step is accomplished within 5…13 RC time constants (depending on precision)
- Precise tracking of a high-frequency continuous time input signal tends to impose more stringent requirements
  - Number to remember: ~1% attenuation error at Nyquist ($f_{in}=f_s/2$) for N ~10
- In applications where attenuation is tolerable, the RC time constant requirements then tend to follow from the distortion specs
  - The larger the attenuation, the larger the instantaneous voltage drop across the (weakly nonlinear) MOSFET → undesired harmonics
    - More later

Thermal Noise



- Sample values $V_{out}(n)$ correspond to instantaneous values of the track mode noise process

$$\overline{v_{out,tot}^2} = \int_0^\infty 4kTR \cdot \left| \frac{1}{1 + j2\pi f \cdot RC} \right|^2 df = \frac{kT}{C}$$

Implications of kT/C Noise

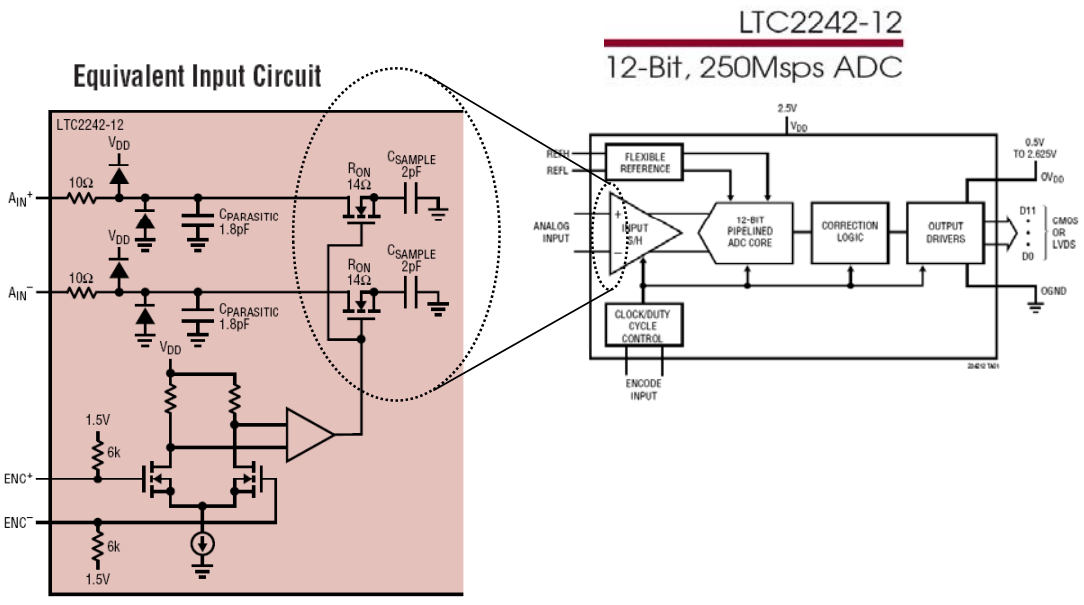
- Example: suppose we make the kT/C noise equal to the quantization noise of a B-bit ADC

$$\frac{kT}{C} = \frac{\Delta^2}{12}, \quad \Delta = \frac{V_{FS}}{2^B} \Rightarrow \quad C = 12kT\left(\frac{2^B}{V_{FS}}\right)^2$$

- For a given B, both C and R (via N on slide 10) are fully determined
- Example numbers for V_{FS}=1V and f_s=100MHz:

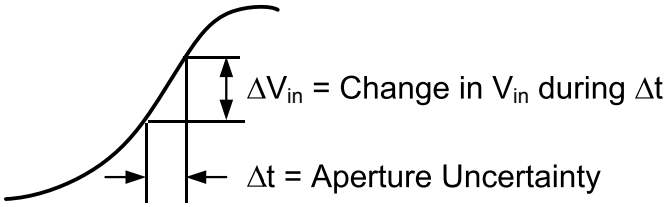
B	C [pF]	R [Ω]
8	0.003	246,057
10	0.052	12,582
12	0.834	665
14	13.3	36
16	213	1.99
18	3,416	0.11

Commercial Example



Aperture Uncertainty

- In any sampling circuit, electronic noise causes random timing variations in the actual sampling clock edge
  - Adds "noise" to samples, especially if dV_{in}/dt is large



$$\Delta V_{in} \cong \frac{dV_{in}}{dt} \cdot \Delta t$$

- Analysis
  - Consider sine wave input signal
  - Assume Δt is random with zero mean and standard deviation σ_t

Analysis

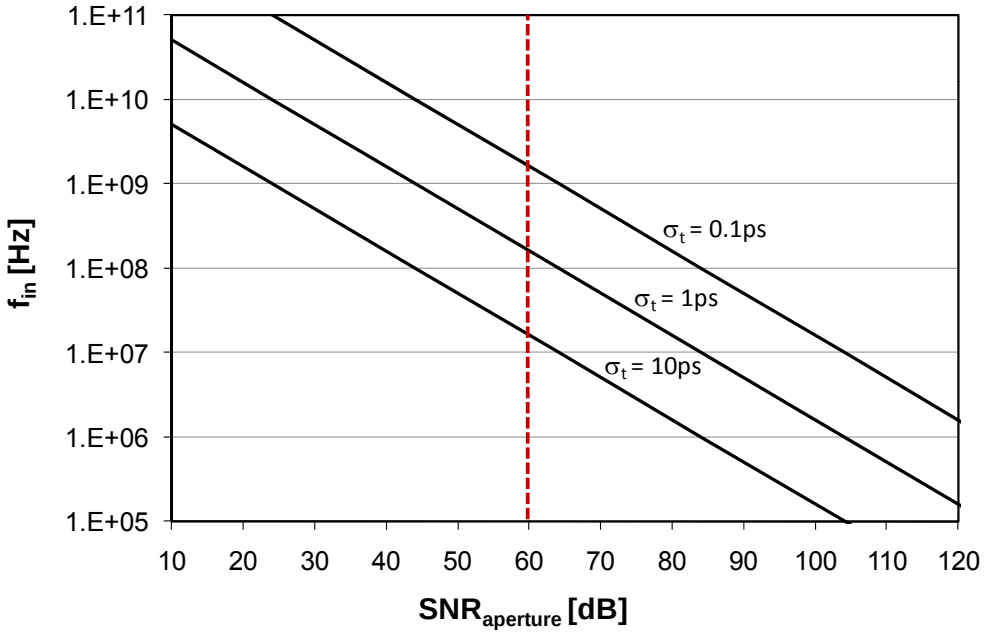
$$E\left\{\Delta V_{in}^2\right\} \cong E\left\{\left(\frac{dV_{in}}{dt}\right)^2 \cdot \Delta t^2\right\} = E\left\{\left(\frac{dV_{in}}{dt}\right)^2\right\} \cdot E\left\{\Delta t^2\right\}$$

$$\cong E\left\{\left(\frac{d}{dt} A\cos\left[2\pi \cdot f_{in} \cdot t\right]\right)^2\right\} \cdot \sigma_t^2 \cong \frac{1}{2}\left(2\pi \cdot A \cdot f_{in}\right)^2 \cdot \sigma_t^2$$

$$SNR_{aperture}[dB] \cong 10 \cdot \log\left[\frac{\frac{1}{2}A^2}{\frac{1}{2}\left(2\pi \cdot A \cdot f_{in} \cdot \sigma_t\right)^2}\right] = 20 \cdot \log\left[\frac{1}{2\pi \cdot f_{in} \cdot \sigma_t}\right]$$

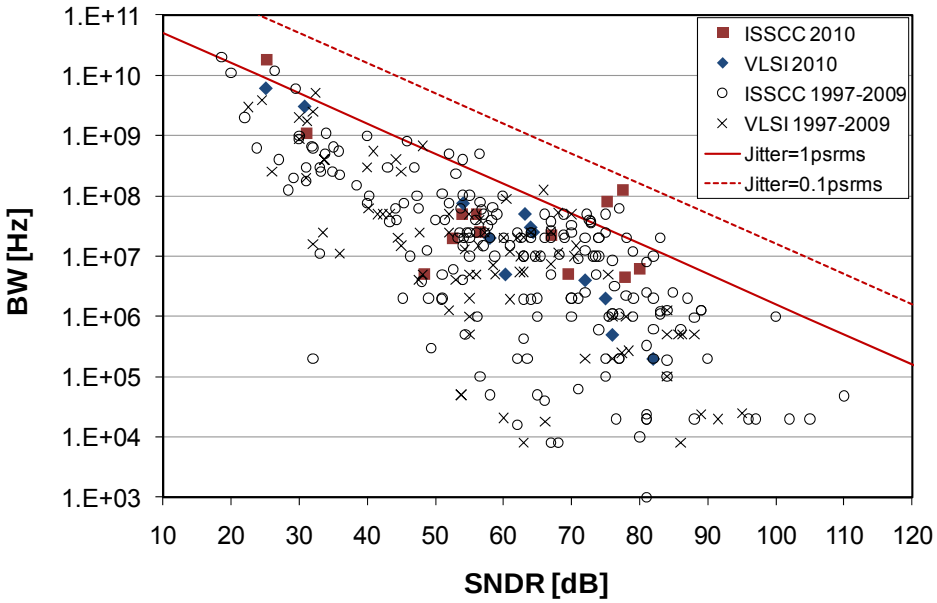
- For an input signal whose power is evenly distributed between 0…f_s/2, the above result improves by 4.8 dB
  - See e.g. [Da Dalt, TCAS1, 9/2002]

Result



ADC Performance Survey (ISSCC & VLSI 97-10)

Data: http://www.stanford.edu/~murmann/adcsurvey.html

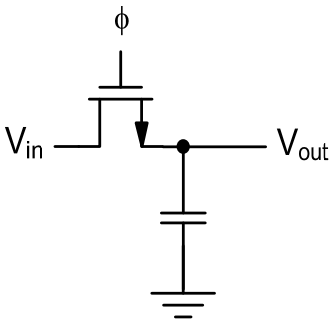


Voltage Dependence of Switch

$$I_{D(triode)} = \mu C_{ox} \frac{W}{L}\left(V_{GS} - V_t - \frac{V_{DS}}{2}\right)V_{DS}$$

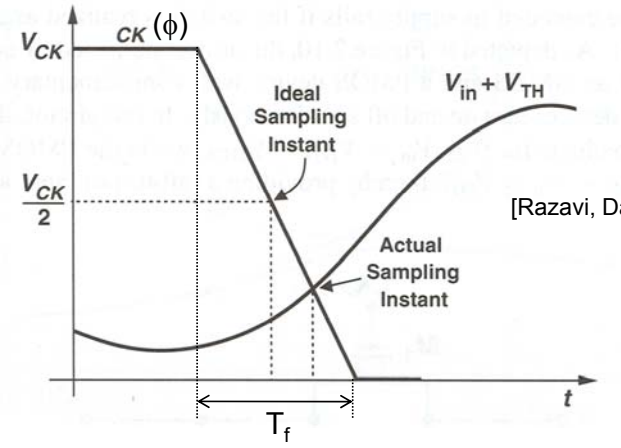
$$R_{ON} \cong \left[\left.\frac{dI_{D(triode)}}{dV_{DS}}\right|_{V_{DS}\to 0}\right]^{-1} = \frac{1}{\mu C_{ox}\frac{W}{L}(V_{GS} - V_t)}$$

$$R_{ON} = \frac{1}{\mu C_{ox}\frac{W}{L}(\phi - V_{in} - V_t)}$$



- Two problems
  - Transistor turn off is signal dependent, occurs when $\phi=V_{in}+V_t$
  - R_{ON} is modulated by V_{in} (assuming e.g. $\phi=V_{DD}$=const.)

Signal Dependent Sampling Instant (1)



[Razavi, Data Conversion System Design, p.17]

- Must make fall time of sampling clock (T_f) much faster than maximum dV_{in}/dt

Signal Dependent Sampling Instant (2)

- Distortion analysis result (see Yu, TCAS II, 2/1999]

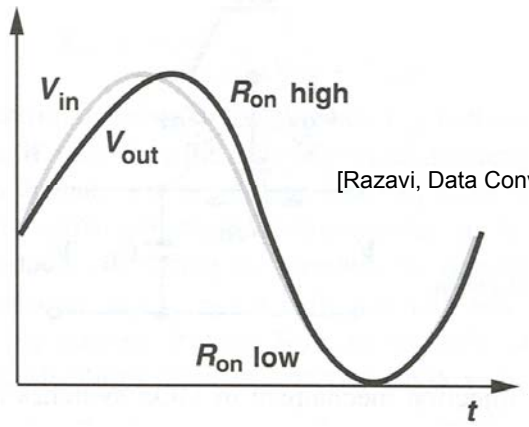
$$HD_3 = \frac{\text{Amplitude of third harmonic}}{\text{Amplitude of fundamental}}$$

$$\cong \frac{3}{8}\left(\frac{A}{V_{CK}}\omega T_f\right)^2$$

- Example: $V_{CK} = 1.8V$, $A = 0.5V$, $T_f = 100ps$, $\omega = 2\pi 100MHz$

$$HD_3 \cong \frac{3}{8}\left(\frac{0.5}{1.8}\cdot 2\pi 100\cdot 10^6\cdot 100\cdot 10^{-12}\right)^2 = -79dB$$

Track Mode Nonlinearity



[Razavi, Data Conversion System Design, p.16]

- Output tracks well when input voltage is low
  - Gets distorted when voltage is high due to increase in R_{ON}

Analysis

$$I_D \cong K\left(V_{GS} - V_t\right)V_{DS} - \frac{K}{2}V_{DS}^2$$

$$C\frac{dV_{out}}{dt} = K\left(\phi - V_{out} - V_t\right)\left(V_{in} - V_{out}\right) - \frac{K}{2}\left(V_{in} - V_{out}\right)^2$$

- "All" we need to do is solve the above differential equation…
- Can use Volterra Series analysis
  - General method that allows us to calculate the frequency domain response of nonlinear circuits with memory
  - See e.g. Stanford EE214
- Luckily someone has already done this for us
  - See [Yu, TCAS II, *2*/1999]

Result

$$HD_3 = \frac{\text{Amplitude of third harmonic}}{\text{Amplitude of fundamental}}$$

$$\cong \frac{1}{4}\left(\frac{A}{V_{GS}-V_t}\right)^2 \cdot \omega\tau = \frac{1}{4}\left(\frac{A}{V_{GS}-V_t}\right)^2 \frac{f_{in}}{f_s} \cdot \frac{\pi}{N}$$

- V_{GS} is the "quiescent point" value of the gate-source voltage; i.e. in the zero crossing of the sine input
- For low distortion
  - Make amplitude smaller than V_{GS}-V_t
    - Low swing → bad for SNR
  - Make $1/\tau$ much larger than ω (input frequency)
    - Big switch → may cost lots of power to drive, comes with large parasitic capacitances

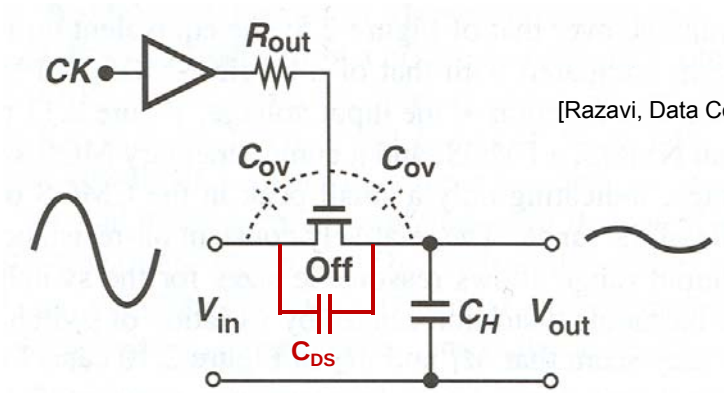
Numerical Example

- Parameters
  - $V_{DD} = V_{CK} = 1.8V$
  - Signal is centered about $V_{DD}/2 = 0.9V$
  - V_{GS}-V_t = 1.8V-0.9V-0.45V = 0.45V
  - A = 0.2V
  - $N = 0.5T_s/\tau = 10$
  - $f_{in} = f_s/2$

$$HD_3 \cong \frac{1}{4}\left(\frac{0.2}{0.45}\right)^2 \frac{1}{2} \cdot \frac{\pi}{10} = -42\text{dB}$$

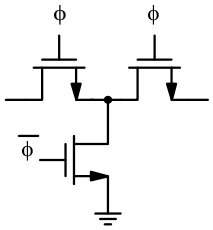
- Not all that great…

Hold Mode Feedthrough



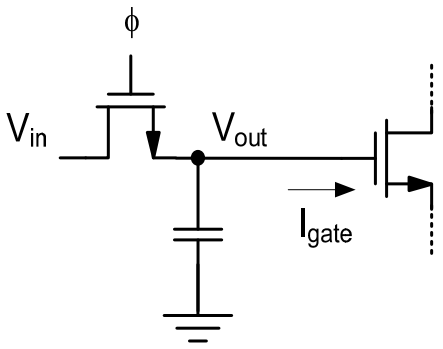
[Razavi, Data Conversion System Design, p.17]

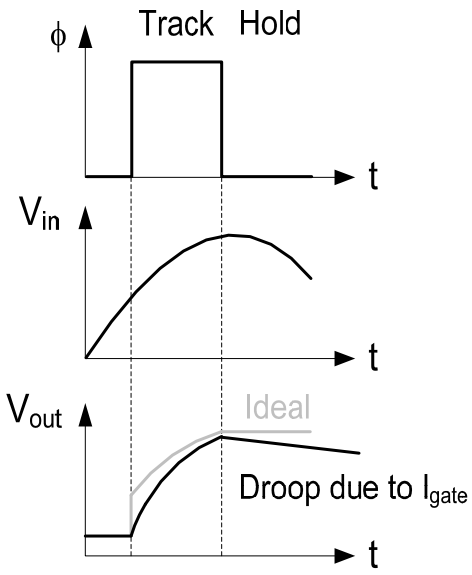
- Want to make R_{out} as small as possible
- Consider using a "T-switch" when hold-mode feedthrough is a problem



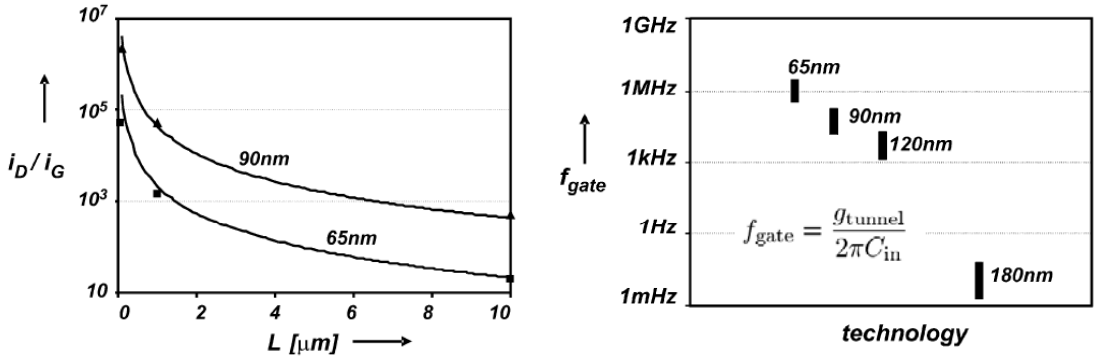
Hold Mode Leakage

Example:





Gate Leakage Data

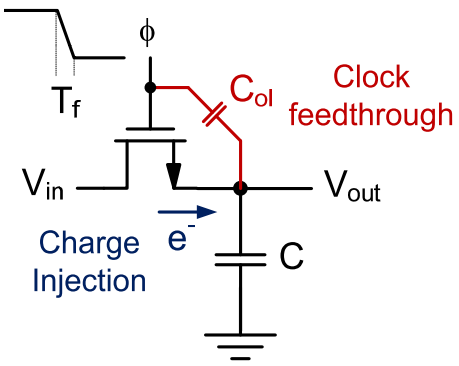


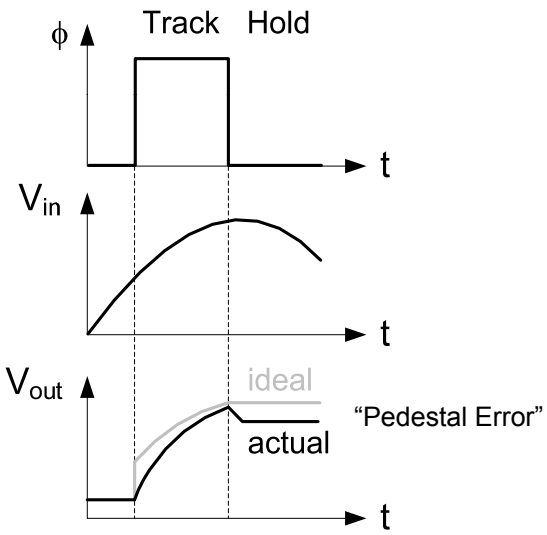
A. Annema, et al., "Analog circuits in ultra-deep-submicron CMOS," *IEEE J. Solid-State Circuits,* pp. 132-143, Jan. 2005.

$$\frac{dv_C}{dt} \approx -\gamma_{dv\,dt} \cdot f_{\text{gate}} \left[\frac{V}{s}\right] \quad \text{with } \gamma_{dv\,dt} \approx 1\text{ V}.$$

- In 65nm CMOS, gate capacitance droop rate is ~1V/μs (!)
- Issue is solved with high-k dielectrics in post-65nm technologies

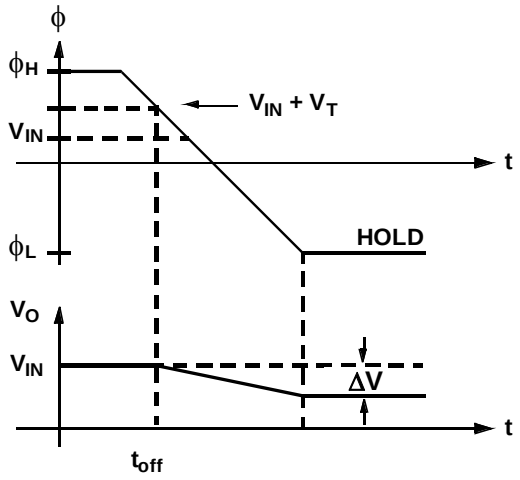
Charge Injection and Clock Feedthrough





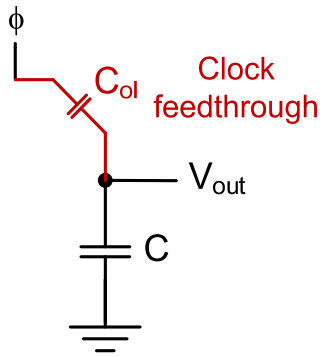
- Analyze two extreme cases
  - Very large T_f ("slow gating")
  - Very small T_f ("fast gating")

Slow Gating



- All channel charge has disappeared by t_{off} without introducing error; it is absorbed by the input source

Slow Gating Model for t > t_{off}



$$V_{out} = V_{in} - \Delta V_{out}$$

$$V_{out} = V_{in} - \frac{C_{ol}}{C_{ol}+C}\left(V_{in} + V_t - \phi_L\right) = V_{in}\left(1+\varepsilon\right) + V_{os}$$

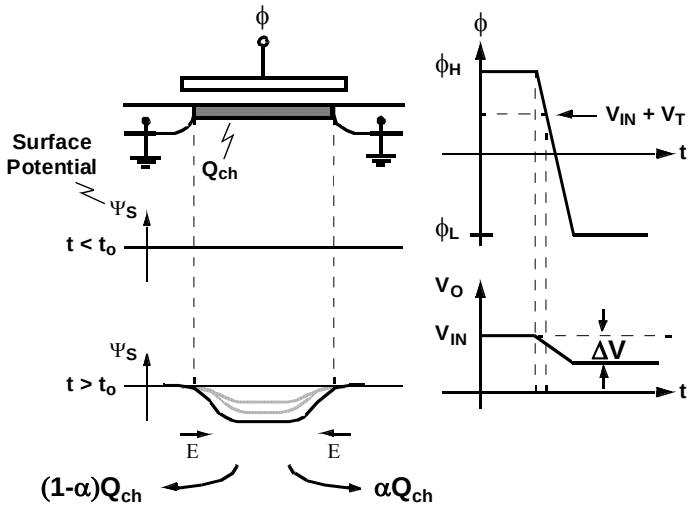
$$\varepsilon = -\frac{C_{ol}}{C_{ol}+C} \qquad V_{os} = -\frac{C_{ol}}{C_{ol}+C}\left(V_t - \phi_L\right)$$

Gain Error Offset Error

- Example: C=1pF, ϕ_L=0V, V_t=0.45V, W=20μm, C_{ol}'=0.1fF/μm, C_{ol}=2fF

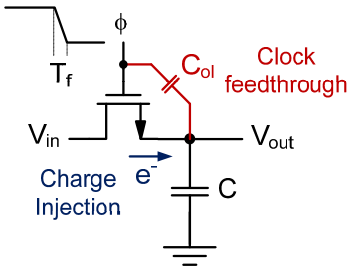
$$\varepsilon = -0.2\% \qquad V_{os} = -0.9\text{mV}$$

Fast Gating



- Channel charge cannot change instantaneously
- Resulting surface potential decays via charge flow to source and drain
- Charge divides between source and drain depending on impedances loading these nodes

Fast Gating Model for $t > t_{off}$



$$V_{out} = V_{in} - \Delta V_{out} = V_{in}(1+\varepsilon) + V_{os}$$

$$V_{out} = V_{in} - \frac{C_{ol}}{C_{ol}+C}(\phi_H - \phi_L) + \frac{1}{2}\frac{Q_{ch}}{C}$$

Assuming 50/50 charge split

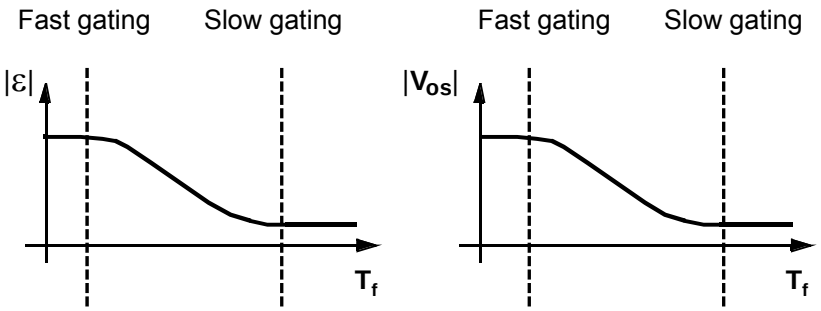
$$Q_{ch} = -WLC_{ox}\left[\phi_H - V_{in} - V_t\right]$$

$$\varepsilon = \frac{1}{2}\frac{WLC_{ox}}{C} \qquad V_{os} = -\frac{C_{ol}}{C_{ol}+C}(\phi_H - \phi_L) - \frac{1}{2}\frac{WLC_{ox}}{C}(\phi_H - V_t)$$

- Example: C=1pF, ϕ_H-ϕ_L=1.8V, V_t=0.45V, W=20μm, LC_{ox}=2fF/μm C_{ol}'=0.1fF/μm, C_{ol}=2fF

$$\varepsilon = +2\% \qquad V_{os} = -30.6\text{mV}$$

Transition Fast/Slow Gating



- $|\varepsilon|$ and $|V_{os}|$ decrease as the fall time of ϕ (T_f) increases and approach the limit case of slow gating
- Unfortunately, high-speed switched capacitor circuits tend to operate in fast gating regime

Speed/Accuracy Trade Off

$$\left.\begin{aligned} \Delta V \cong \frac{1}{2}\frac{Q_{ch}}{C} \qquad & BW \cong 1/\tau = 1/(RC) \\ R \cong \frac{1}{\mu C_{ox}\frac{W}{L}(V_{GS}-V_t)} = \frac{L^2}{\mu Q_{ch}} \qquad & \end{aligned}\right\} \quad \frac{\Delta V}{BW} \cong \frac{L^2}{\mu}$$

- Charge injection error to speed ratio benefits from shorter channels and increases in mobility (e.g. due to strain)

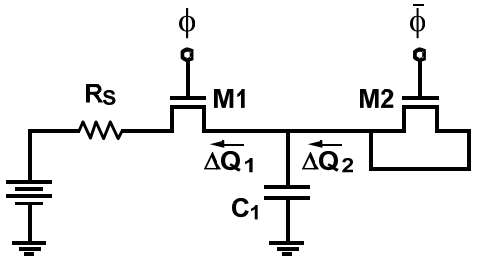
Outline

- Elementary track-and-hold circuit and its nonidealities
- First order improvements to elementary track-and-hold
- Advanced techniques
  - Clock bootstrapping
  - Bottom plate sampling
- Settling and noise analysis in charge-redistribution track-and-hold circuit
- Noise simulation example

Improvements

- Charge cancellation
  - Try to cancel channel charge by injecting a charge packet with opposite sign
- Differential sampling
  - Use a differential circuit to suppress offset
- CMOS switch
  - Try to balance the nonidealities of NMOS device with a parallel PMOS

Charge Cancellation



$$\Delta Q_1 = 0.5Q_{ch1} + Q_{ol1}$$

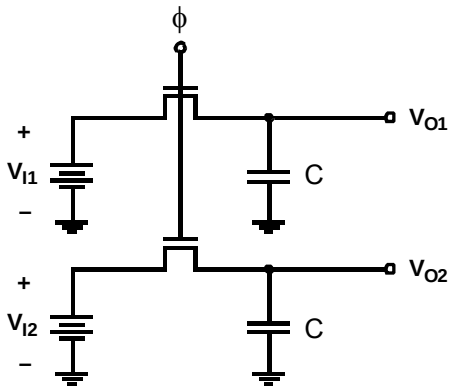
$$\Delta Q_2 \cong Q_{ch2} + 2Q_{ol2} \cong 0.5Q_{ch1} + Q_{ol1}$$

$$\Delta Q_1 - \Delta Q_2 \cong 0$$

[Eichenberger and Guggenbűhl, JSSC 8/89]

- Cancellation is never perfect, since channel charge of M1 will not exactly split 50/50
  - E.g. if R_s is very small, most of M1's channel charge will flow toward the input voltage source
- Not a precision technique, just an attempt to do a partial clean-up

Differential Sampling (1)



$$V_{ID} = V_{I1} - V_{I2} \qquad V_{OD} = V_{O1} - V_{O2}$$

$$V_{IC} = \frac{V_{I1} + V_{I2}}{2} \qquad V_{OC} = \frac{V_{O1} + V_{O2}}{2}$$

$$V_{O1} = (1+\varepsilon_1)V_{I1} + V_{OS1}$$

$$V_{O2} = (1+\varepsilon_2)V_{I2} + V_{OS2}$$

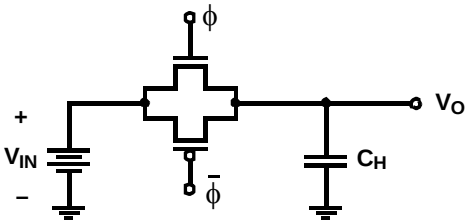
$$V_{OD} = \left(1+\frac{\varepsilon_1+\varepsilon_2}{2}\right)V_{ID} + (\varepsilon_1-\varepsilon_2)V_{IC} + (V_{OS1}-V_{OS2}) \cong \left(1+\frac{\varepsilon_1+\varepsilon_2}{2}\right)V_{ID}$$

$$V_{OC} = \left(\frac{\varepsilon_1-\varepsilon_2}{4}\right)V_{ID} + \left(1+\frac{\varepsilon_1+\varepsilon_2}{2}\right)V_{IC} + \left(\frac{V_{OS1}+V_{OS2}}{2}\right) \cong \left(1+\frac{\varepsilon_1+\varepsilon_2}{2}\right)V_{IC} + \left(\frac{V_{OS1}+V_{OS2}}{2}\right)$$

Differential Sampling (2)

- Assuming good matching between the two half circuits, we have
  - Small residual offset in V_{OD}
  - Good rejection of coupling noise, supply noise, …
  - Small common-mode to differential-mode gain
- Unfortunately, V_{OD} has essentially same gain error as the basic single ended half circuit
- This also means that there will be nonlinear terms
  - Out simplistic analysis assumed that the channel charge is linearly related to Vin
  - This is true only to first order (consider e.g. backgate effect)
    - Expect to see nonlinear distortion along with gain error

CMOS Switch



$$Q_{chn} \cong -W_n L_n C_{ox}\left(\phi_H - V_{IN} - V_{tn}\right)$$

$$Q_{chp} \cong W_p L_p C_{ox}\left(V_{IN} - \phi_L - \left|V_{tp}\right|\right)$$

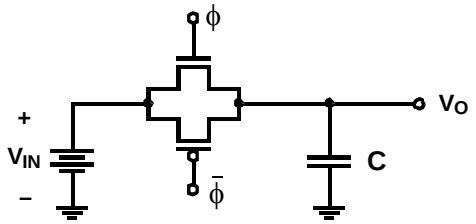
- Assuming fast gating, 50/50 charge split and $W_n L_n = W_p L_p$

$$\Delta V_o \cong \frac{\frac{1}{2}Q_{chn} + \frac{1}{2}Q_{chp}}{C} = \frac{C_{ox}}{C}\left(V_{IN} - \frac{\phi_H - \phi_L}{2} + \frac{V_{tn} - \left|V_{tp}\right|}{2}\right)$$

- Charges fully cancel e.g. for $V_{IN} = (\phi_H - \phi_L)/2 = V_{DD}/2$, and $V_{tn} = |V_{tp}|$, but there is still signal dependent residual injection

On Resistance of CMOS Switch

- At least in principle, adding a PMOS can also help with the problem of signal dependent R_{on} in track mode
  - For increasing V_{IN}, NMOS resistance goes up, PMOS resistance goes down



$$R \cong \frac{1}{\mu_n C_{ox}\left[\frac{W}{L}\right]_n \left(V_{GSn} - V_{tn}\right)} \Bigg\| \frac{1}{\mu_p C_{ox}\left[\frac{W}{L}\right]_p \left(\left|V_{GSp}\right| - \left|V_{tp}\right|\right)}$$

Analysis

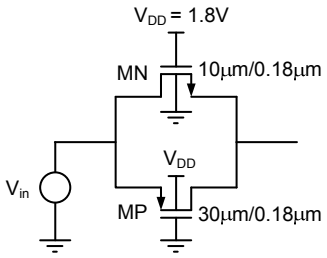
$$R \cong \frac{1}{\mu_n C_{ox}\left[\frac{W}{L}\right]_n \left(V_{GSn} - V_{tn}\right)} \Bigg\| \frac{1}{\mu_p C_{ox}\left[\frac{W}{L}\right]_p \left(\left|V_{GSp}\right| - \left|V_{tp}\right|\right)}$$

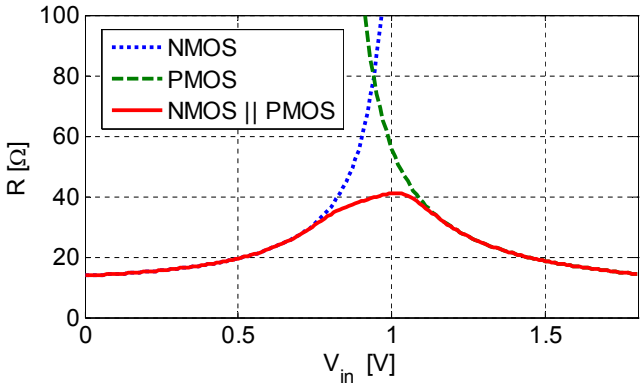
$$R \cong \frac{1}{\mu_n C_{ox}\left[\frac{W}{L}\right]_n \left(V_{DD} - V_{tn}\right) - \left(\mu_n C_{ox}\left[\frac{W}{L}\right]_n - \mu_p C_{ox}\left[\frac{W}{L}\right]_p\right) v_{in} - \mu_p C_{ox}\left[\frac{W}{L}\right]_p \left|V_{tp}\right|}$$

$$R \cong \frac{1}{\mu_n C_{ox}\left[\frac{W}{L}\right]_n \left(V_{DD} - V_{tn} - \left|V_{tp}\right|\right)} \quad \text{if} \quad \mu_n\left[\frac{W}{L}\right]_n = \mu_p\left[\frac{W}{L}\right]_p$$

- Independent of V_{in} → too good to be true!
- Missing factors
  - Body effect
  - Short channel effects

Real CMOS Switch



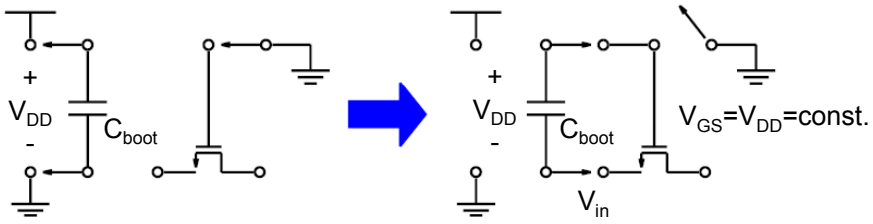


- Design
  - Size P/N ratio to minimize change in R over input range
  - Size P and N simultaneously to meet distortion specs
- PMOS brings limited benefit unless the input signal range is large or centered near V_{DD}

Outline

- Elementary track-and-hold circuit and its nonidealities
- First order improvements to elementary track-and-hold
- Advanced techniques
  - Clock bootstrapping
  - Bottom plate sampling
- Settling and noise analysis in charge-redistribution track-and-hold circuit
- Noise simulation example

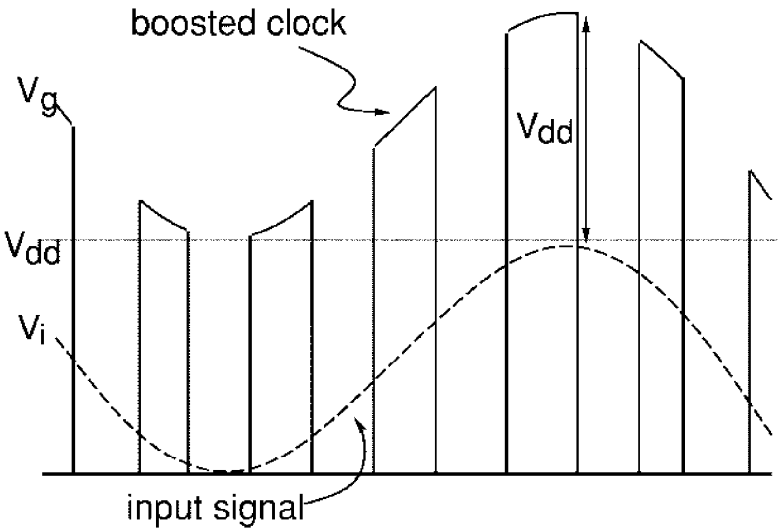
Clock Bootstrapping



A. Abo, "Design for Reliability of Low-voltage, Switched-capacitor Circuits," PhD Thesis, UC Berkeley, 1999.

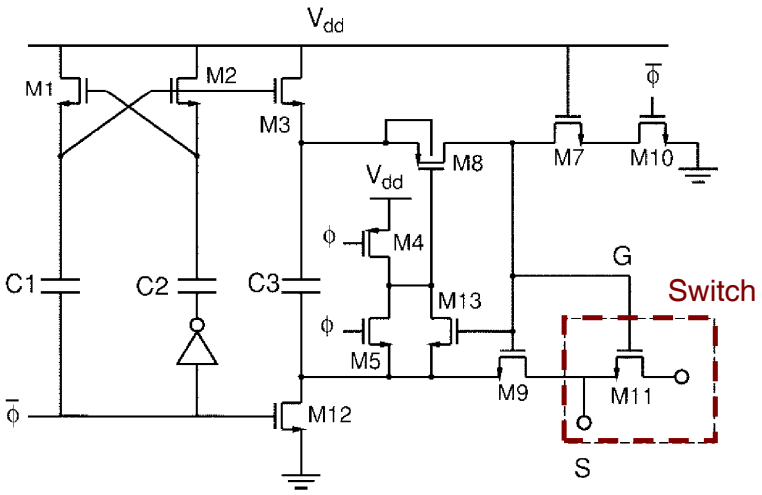
- Phase 1
  - C_{boot} is precharged to V_{DD}
  - Sampling switch is off
- Phase 2
  - Sampling switch is on with $V_{GS}=V_{DD}$=const.
  - To first order, both R_{on} and channel charge are signal independent

Waveforms



A. Abo, "Design for Reliability of Low-voltage, Switched-capacitor Circuits," PhD Thesis, UC Berkeley, 1999.

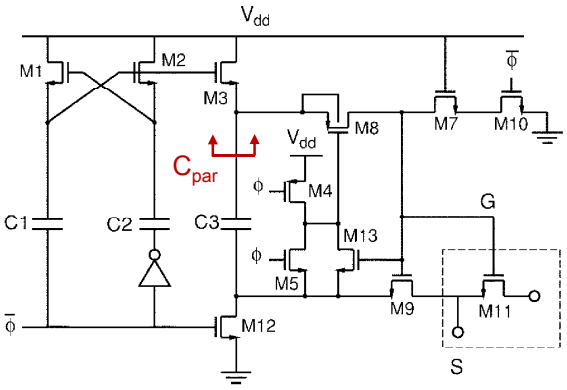
Circuit Implementation



A. Abo et al., "A 1.5-V, 10-bit, 14.3-MS/s CMOS Pipeline Analog-to-Digital Converter," IEEE J. Solid-State Circuits, pp. 599, May 1999

Limitations

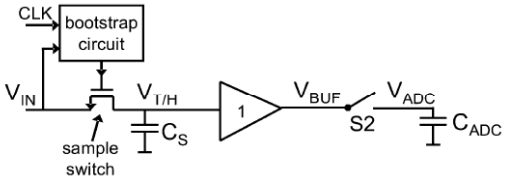
- Efficacy of bootstrap circuit is reduced by
  - Body effect
  - Parasitic capacitance between at top plate of C3



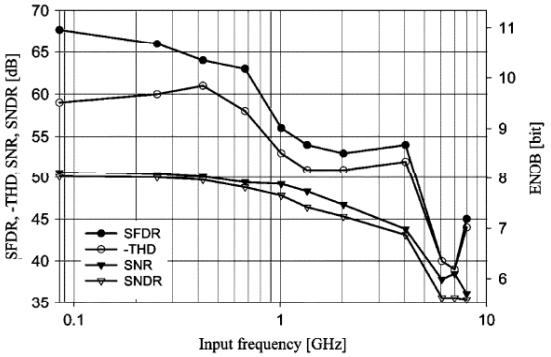
$$R \cong \frac{1}{\mu_n C_{ox}\left[\frac{W}{L}\right]_n\left(\frac{C_{boot}}{C_{boot}+C_{par}}V_{DD}-\frac{C_{par}}{C_{boot}+C_{par}}V_{in}-\underbrace{V_{tn}[V_{in}]}_{\text{Backgate effect}}\right)}$$

Performance of Bootstrapped Samplers

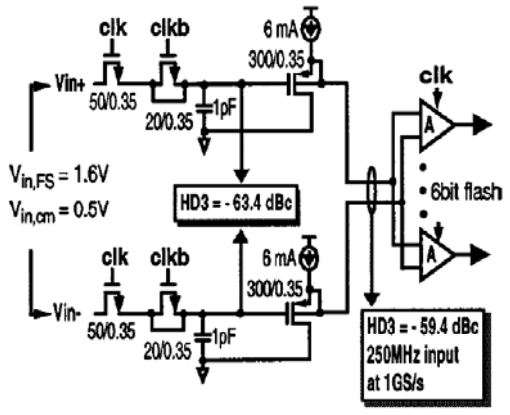
- Bootstrapped “top plate” sampling (as opposed to “bottom plate” – more later) tends to work very well up to ~10bit resolution
- Example



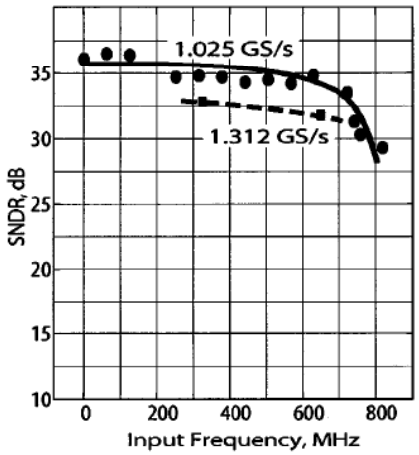
[Louwsma, JSSC 4/2008]



High-Speed Example without Bootstrap



[Choi and Abidi, JSSC 12/2001]

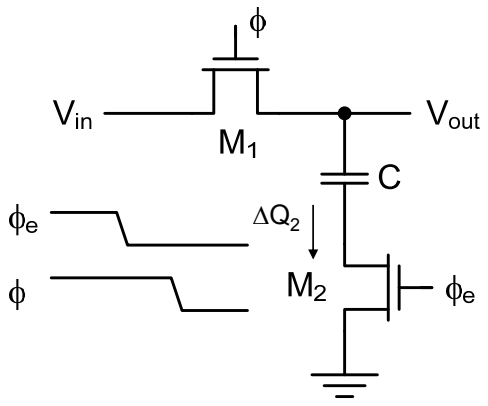


- For lower resolution applications, it can be OK to drop the bootstrap circuit

Bottom Plate Sampling

- Used to eliminate input-dependent charge injection error.
- Basic idea
  - Sample signal at the "grounded" side of the capacitor to achieve signal independent sampling
- References
  - D. J. Allstot and W. C. Black, Jr., “Technological Design Considerations for Monolithic MOS Switched-Capacitor Filtering Systems,” Proc. IEEE, pp. 967-986, Aug. 1983.
  - K.-L. Lee and R. G. Meyer, “Low-Distortion Switched-Capacitor Filter Design Techniques,” IEEE J. Solid-State Circuits, pp. 1103-1113, Dec. 1985.
- First look at single ended half circuit for simplicity

Bottom Plate Sampling Analysis (1)



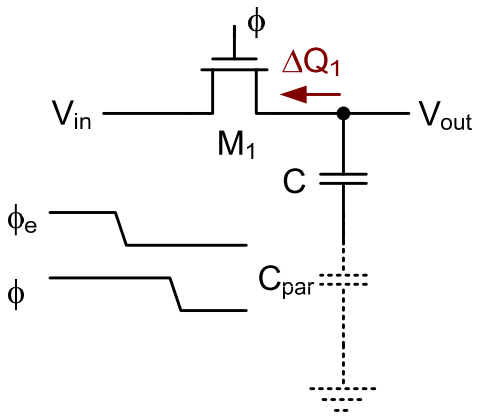
- Turn M_2 off "slightly" before M_1
  - Typically a few hundred ps delay between falling edges of ϕ_e and ϕ
- During turn off, M_2 injects charge

$$\Delta Q_2 \cong \frac{1}{2} WLC_{ox}\left(\phi_H - V_{tn}\right)$$

- To first order, the charge injected by M_2 is signal independent
- Voltage across C

$$V_C = V_{in} + \frac{\Delta Q_2}{C}$$

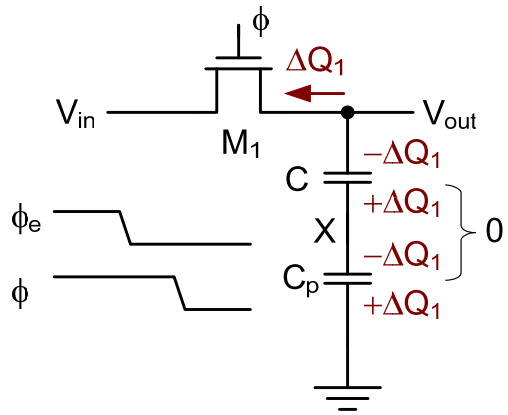
Bottom Plate Sampling Analysis (2)



$$\Delta Q_1 \cong \frac{1}{2} W L C_{ox} \left(\phi_H - V_{in} - V_{tn} \right)$$

- Next, turn off M1
- M1 will inject signal dependent charge onto the series combination of C and the parasitic capacitance at its bottom plate (C_{par})
- Looks like, this is not much different from the conventional top-plate sampling?
  - But wait…

Bottom Plate Sampling Analysis (3)

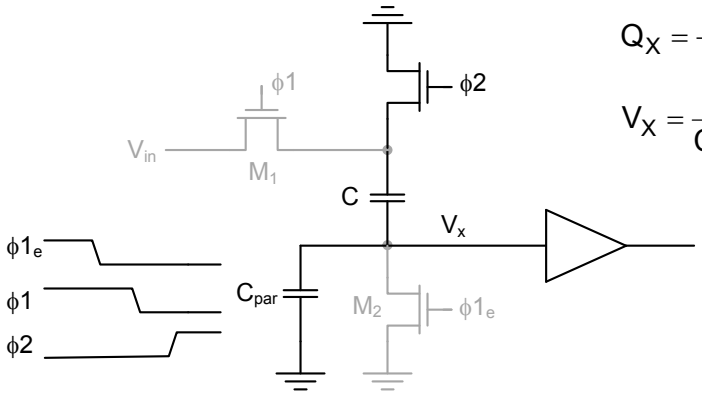


$$Q_X = -CV_{in} - \underbrace{\Delta Q_2}$$

Charge injected by M2
(Signal independent)

- Interesting observation
  - Even though M1 injects some charge, the total charge at node X cannot change!
- Idea
  - Process total **charge** at node X instead of looking at voltage across C
- The charge can be processed in two ways
  - Open-loop
  - Closed-loop (charge redistribution)

Open-Loop Charge Processing



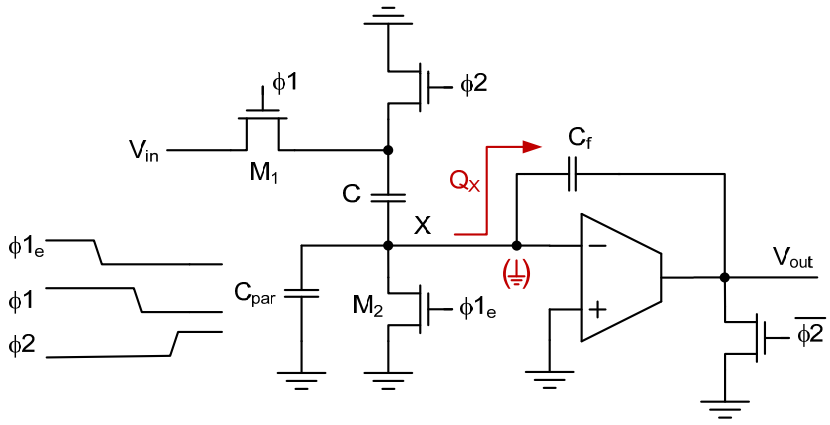
$$Q_X = -CV_{in} - \Delta Q_2$$

$$V_X = \frac{Q_X}{C + C_p} = -V_{in}\frac{C}{C + C_p} - \frac{\Delta Q_2}{C + C_p}$$

(no term due to signal dependent charge!)

- Remaining drawback
  - C_{par} (and buffer input capacitance) is usually weakly nonlinear and will introduce some harmonic distortion

Closed-Loop Charge Processing



- Amplifier forces voltage at node X to "zero"
  - Means that charge at node X must redistribute onto feedback capacitor C_f

Charge Conservation Analysis

Charge at node X during $\phi 1$: $Q_{X1} = -CV_{in} - \Delta Q_2 + 0 \cdot C_f$

Charge at node X during $\phi 2$: $Q_{X2} = -C_f V_{out}$

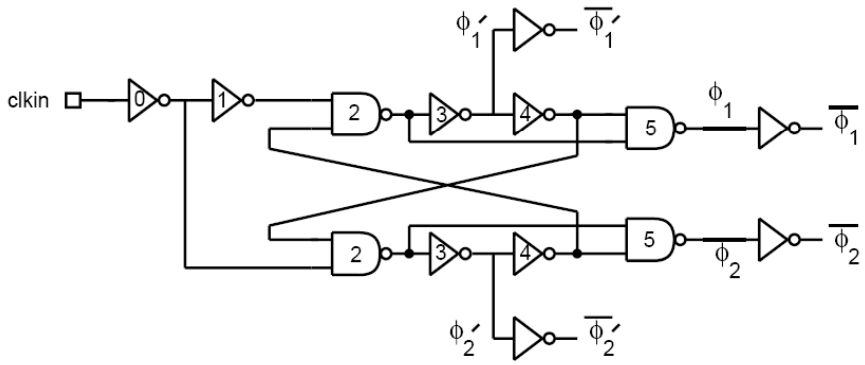
Charge Conservation:

$$Q_{X1} = Q_{x2}$$
$$-CV_{in} - \Delta Q_2 = -C_f V_{out}$$

$$\therefore V_{out} = \frac{C}{C_f} V_{in} + \frac{\Delta Q_2}{C_f}$$

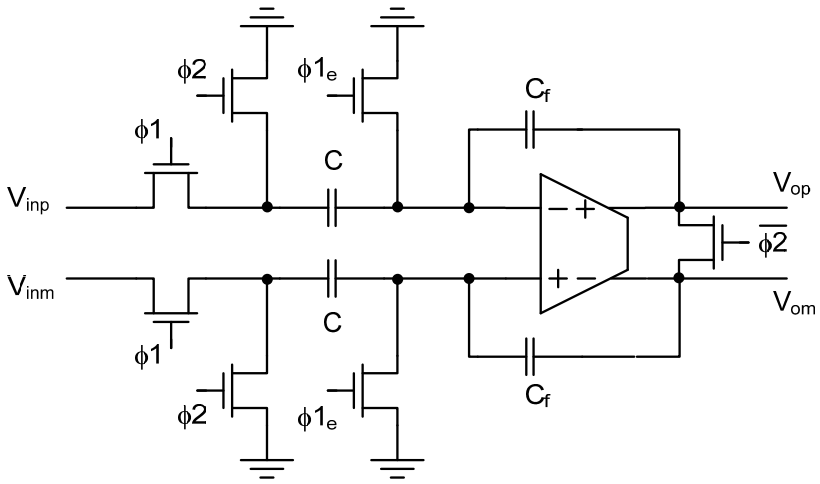
- Offset term due to signal independent injection from M2 can be easily removed using a differential architecture

Clock Generation



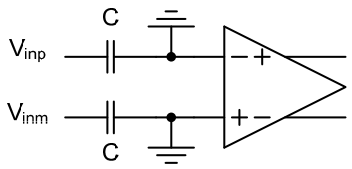
[A. Abo, "Design for Reliability of Low-voltage, Switched-capacitor Circuits," PhD Thesis, UC Berkeley, 1999]

Fully Differential Circuit

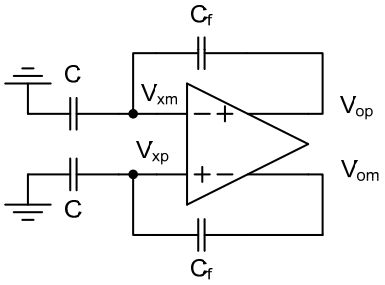


Analysis (1)

During ϕ1



During ϕ2



$$Q_{1m} = -CV_{inp} + \Delta Q$$

$$Q_{1p} = -CV_{inm} + \Delta Q$$

$$Q_{2m} = CV_{xm} - C_f\left(V_{op} - V_{xm}\right)$$

$$Q_{2p} = CV_{xp} - C_f\left(V_{om} - V_{xp}\right)$$

1) $Q_{1m} = Q_{2m}$

2) $Q_{1p} = Q_{2p}$

$$V_{xm} = V_{xp}$$

$$\frac{V_{op} + V_{om}}{2} = V_{oc}$$

Analysis (2)

- Subtracting 1) and 2) yields

$$V_{op} - V_{om} = \frac{C}{C_f}\left(V_{inp} - V_{inm}\right)$$

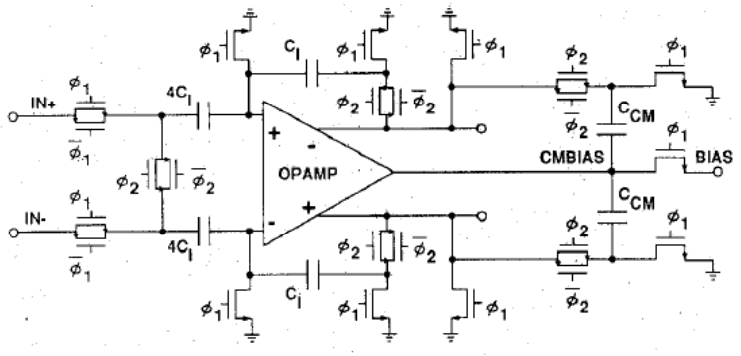
- Adding 1) and 2) yields

$$-C\left(V_{inp} + V_{inm}\right) + 2\Delta Q = \left(C + C_f\right)\left(V_{xp} + V_{xm}\right) - C_f\left(V_{op} + V_{op}\right)$$

$$V_{xc} = \frac{\Delta Q}{C + C_f} + \frac{C_f}{C + C_f} V_{oc} - \frac{C}{C + C_f} V_{ic}$$

- Variations in V_{ic} show up as common mode variations at the amplifier input
  - Need amplifier with good CMRR

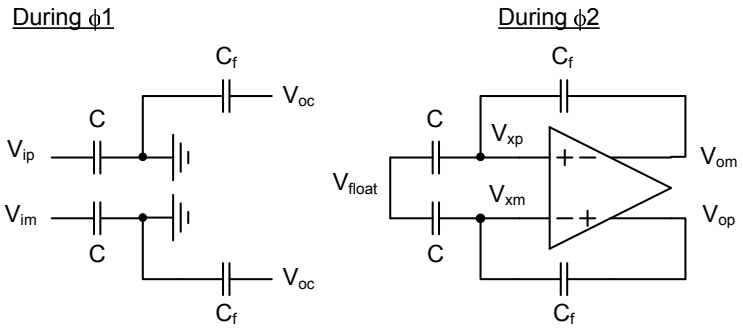
T/H with Common Mode Cancellation



S.H. Lewis & P.R. Gray, "A Pipelined 5 MSample/s 9-bit Analog-to-Digital Converter", IEEE J. Solid-State Circuits, pp. 954-961, Dec. 1987

- Shorting switch allows to re-distribute only differential charge on sampling capacitors
- Common mode at OPAMP input becomes independent of common mode at circuit input terminals (IN+/IN-)
- Original idea: Yen & Gray, JSSC 12/1982

Analysis (1)



- Charge conservation at V_{ip}, V_{im} and V_{float}

$$\left(V_{ip}+V_{im}\right)\cdot C=\left(V_{float}-V_{xp}\right)\cdot C+\left(V_{float}-V_{xm}\right)\cdot C$$

$$V_{ic}=V_{float}-V_{xc}$$

$$V_{float}=V_{ic}+V_{xc}$$

Analysis (2)

- Common mode charge conservation at amplifier inputs

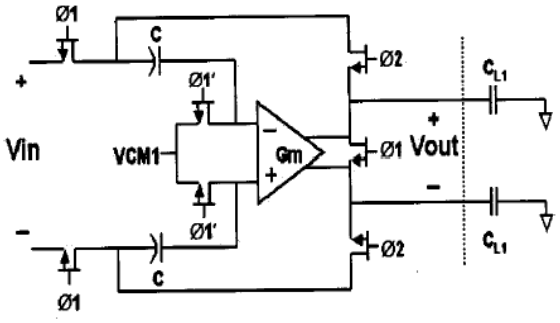
$$-V_{ic}\cdot C-V_{oc}\cdot C_f=-\left(V_{float}-V_{xc}\right)\cdot C-\left(V_{oc}-V_{xc}\right)\cdot C_f$$

$$-V_{ic}\cdot C=-\left(\left[V_{ic}+V_{xc}\right]-V_{xc}\right)\cdot C+V_{xc}\cdot C_f$$

$$0=V_{xc}$$

- Amplifier input common mode (V_{xc}) is independent of
  - Input common mode (V_{ic})
  - Output common mode (V_{oc})

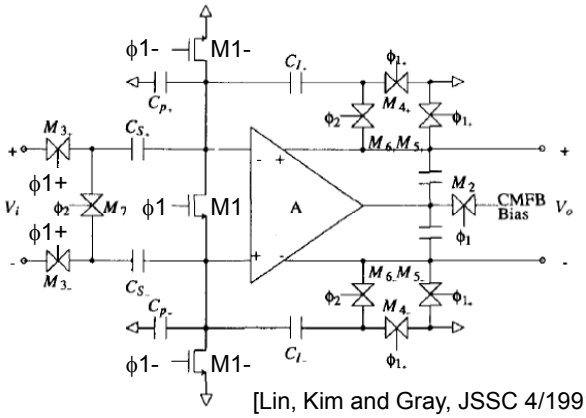
Flip-Around T/H



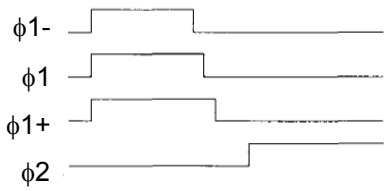
[W. Yang et al., "A 3-V 340-mW 14-b 75-MSample/s CMOS ADC With 85-dB SFDR at Nyquist Input", IEEE J. Solid-State Circuits, pp. 1931-1936, Dec. 2001]

- Sampling caps are "flipped around" OTA and used as feedback capacitors during ϕ_2
- Main advantage: improved feedback factor (lower noise, higher speed)
- Main disadvantage: OTA is subjected to input common mode variations

Sampling Network Design Considerations

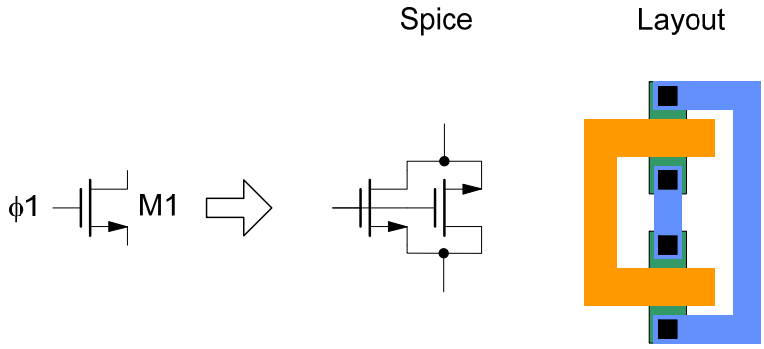


[Lin, Kim and Gray, JSSC 4/1991]



- M1- switches only needed to set common mode; M1 is actual sampling switch
  - Make M1 larger than M1-
- Ideally turn off M1- before M1
  - In practice, usually OK to turn off simultaneously
- In track mode, the total path resistance is R(M3) plus bottom plate switch resistance
  - Since R(M3) is signal dependent, make its resistance small compared to that of bottom plate network

Schematic Entry and Layout of M1



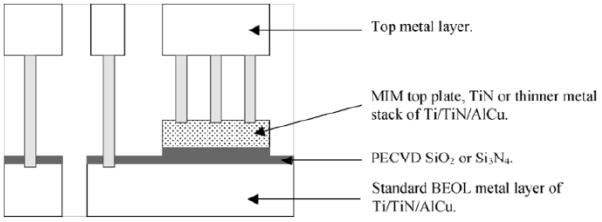
- Use antiparallel devices to implement M1
  - Needed in simulation to guarantee circuit symmetry
    - E.g. BSIM model is not necessarily perfectly symmetric with respect to drain/source!
  - Needed in layout to ensure symmetry in presence of drain/source asymmetry due to processing artifacts

What Ultimately Limits Linearity?

- Track mode nonlinearity due to $R=f(V_{in})$
  - Mitigate using clock bootstrapping and proper partitioning of total path resistance
  - Eventually, bootstrapping falls apart high frequencies, due to parasitics capacitances inside the bootstrap circuit
- Mismatch in half-circuit charge injection due to $R=f(V_{in})$
  - Bottom plate switches in the two half circuits see input dependent impedance; this creates input dependent charge injection mismatch
  - Bootstrapping helps; ultimately limited by backgate effect
  - This effect is often fairly independent of frequency (somewhat dependent on realization of top plate switch)
- In high performance designs, can achieve ~80-100dB linearity up to a few hundreds of MHz

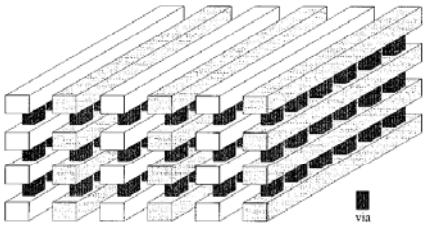
Capacitors

Metal-Insulator-Metal (MIM)



[Ng, Trans. Electron Dev., 7/2005]

Vertical Parallel Plate (VPP)

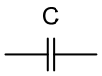


[Aparicio, JSSC 3/2002]

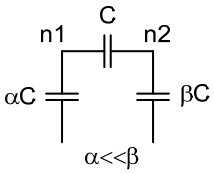
- Typically 1-2 fF/μm^2 (10-20 fF/μm^2 for advanced structures)
  - For 1 fF/μm^2, a 10 pF capacitor occupies ~100μm x 100μm
- Both MIM and VPP capacitors have good electrical properties
  - Mostly worry about parasitic caps
  - Series and parallel resistances are often not a concern

Plate Parasitics

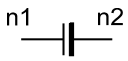
Ideal Capacitor



Typical Integrated Circuit Capacitor

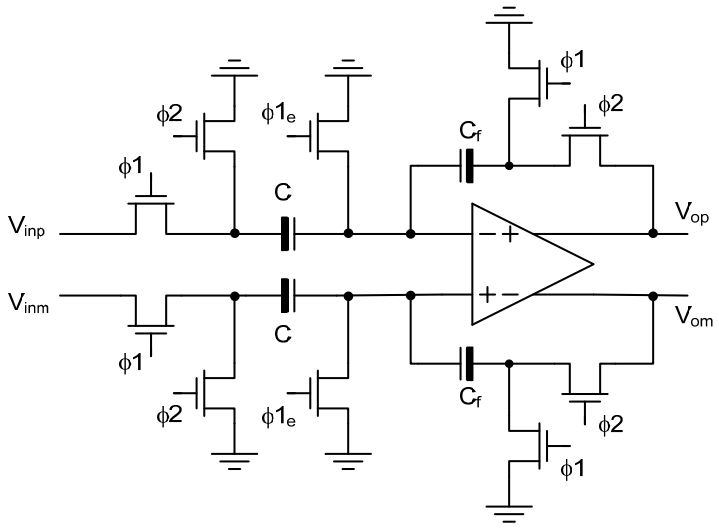


Symbol



- Node n1 is usually the "physical" top plate of the capacitor
  - Makes nomenclature very confusing, since this plate is typically used as the "electrical" bottom plate in a sampling circuit (in the context of "bottom plate sampling")
- Typical values for a MIM capacitor
  - α=1%, β=10%

Proper Connection of Capacitors

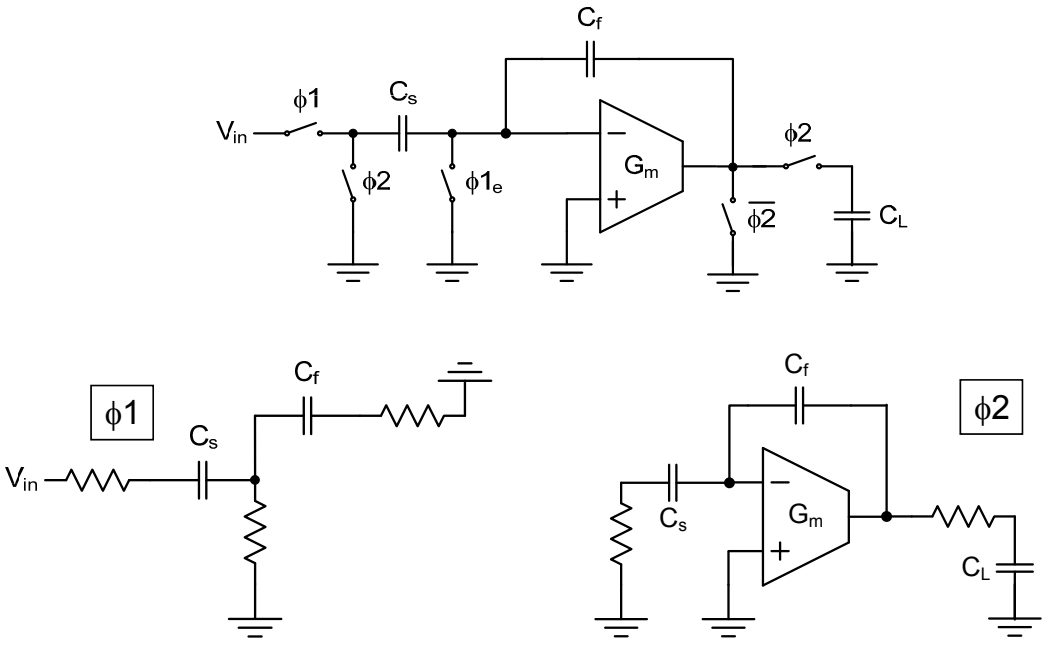


- "Fat plate" is oriented away from virtual ground nodes to avoid reduction of feedback factor and reduce potential noise coupling

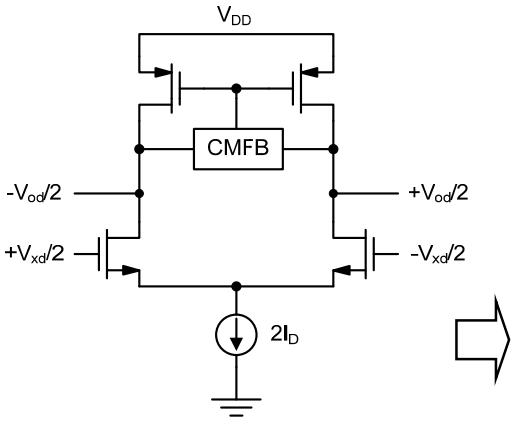
Outline

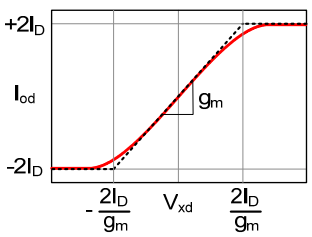
- Elementary track-and-hold circuit and its nonidealities
- First order improvements to elementary track-and-hold
- Advanced techniques
  - Clock bootstrapping
  - Bottom plate sampling
- Settling and noise analysis in charge-redistribution track-and-hold circuit
- Noise simulation example

Settling and Noise Analysis

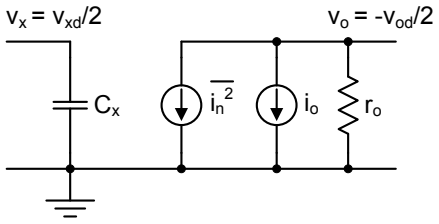


First Order Amplifier Model



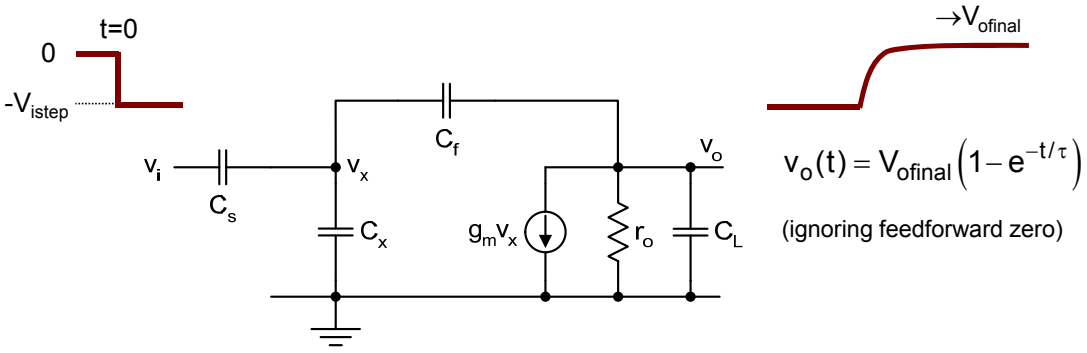


Piecewise linear half-circuit



$$i_o = \begin{cases} g_m v_x & \text{for } |g_m v_x| < I_D \\ I_D \cdot \text{sign}(v_x) & \text{else} \end{cases}$$

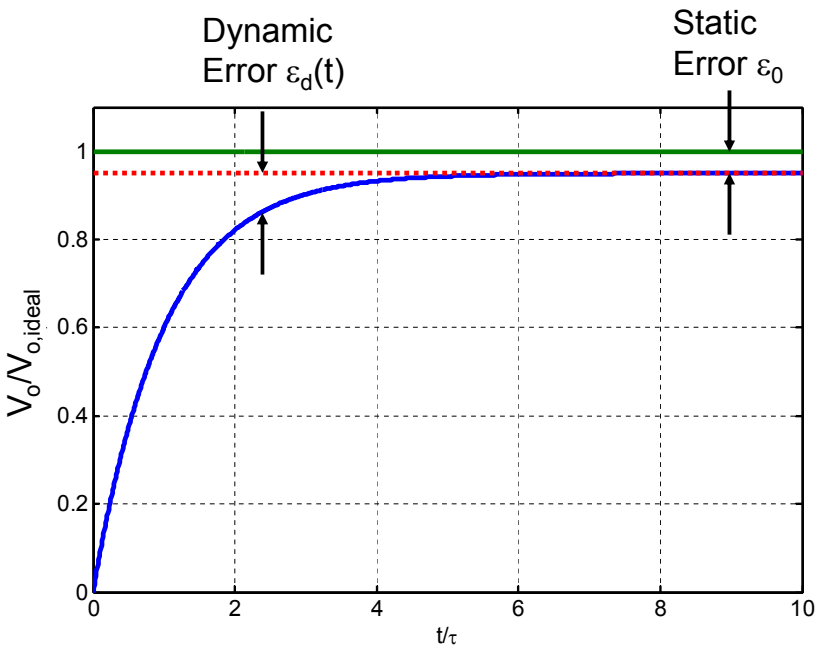
Linear Settling (Small Input Step)



- Important parameter: Return factor or "feedback factor" β

$$\beta = \frac{C_f}{C_f + C_s + C_x}$$

Waveform Detail



Static Settling Error

- Ideal output voltage for $t \to \infty$

$$V_{ofinal,ideal} = V_{istep} \cdot \frac{C_s}{C_f}$$

- Actual output voltage (from detailed analysis)

$$V_{ofinal} = V_{istep} \cdot \frac{C_s}{C_f} \cdot \frac{T_0}{1+T_0} \qquad T_0 = \beta \cdot g_m r_o = \beta \cdot a_{vo}$$

- Define static settling error

$$\varepsilon_0 = \frac{V_{ofinal} - V_{ofinal,ideal}}{V_{ofinal,ideal}} = \frac{\frac{T}{1+T} - 1}{1} = -\frac{1}{1+T} \cong -\frac{1}{T}$$

- Example: T_0=1000 → 0.1% static settling error

Dynamic Settling Error

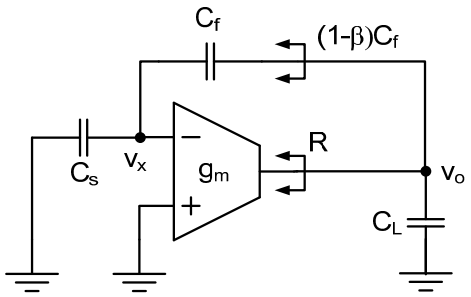
$$\varepsilon_{dynamic}(t) = \frac{v_o(t) - V_{ofinal}}{V_{ofinal}} = \frac{V_{ofinal}\left(1 - e^{-t/\tau}\right) - V_{ofinal}}{V_{ofinal}} = -e^{-t/\tau}$$

$$N = \frac{t_s}{\tau} = -\ln\left(\varepsilon_d\right)$$

$\varepsilon_{dynamic}$	N
1%	4.6
0.1%	6.9
0.01%	9.2

Time Constant

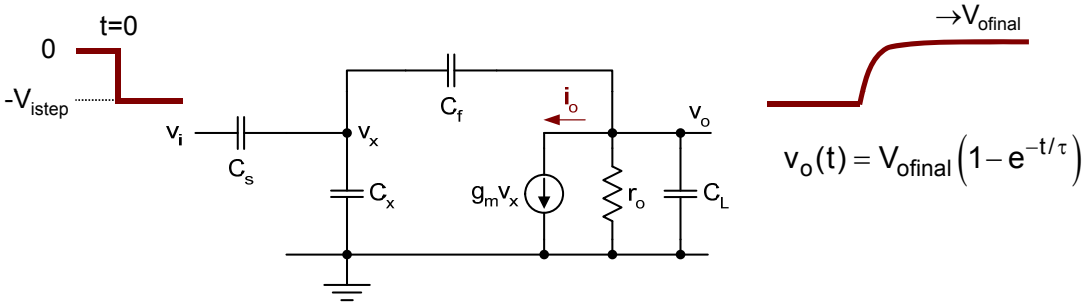
$$\beta = \frac{C_f}{C_f + C_s + C_x}$$



$$R = \frac{1}{\beta g_m} \qquad C_{Ltot} = C_L + (1-\beta)C_f$$

$$\boxed{\tau = \frac{1}{\beta} \cdot \frac{C_{Ltot}}{g_m}}$$

Transconductor Current



- During linear settling, the current delivered by the transconductor is

$$i_o \cong -C_{Ltot}\frac{dv_o(t)}{dt} = -C_{Ltot}\frac{V_{ofinal}}{\tau}e^{-t/\tau}$$

- Peak current occurs at t=0

$$|i_o|_{max} = C_{Ltot}\frac{V_{ofinal}}{\tau}$$

Slewing

- The amplifier can deliver a maximum current of I_D
  - If $|i_o|_{max} > I_D$, slewing occurs

$$|i_o|_{max} = C_{Ltot}\frac{V_{ofinal}}{\tau} > I_D$$

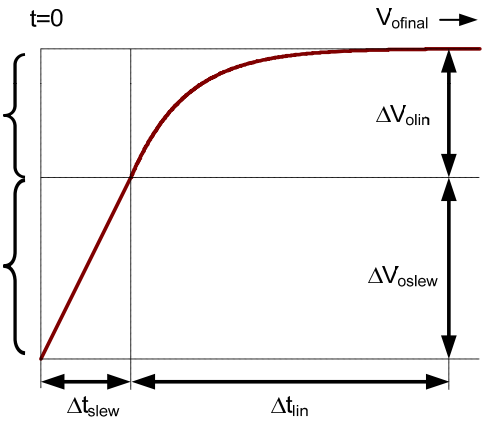
$$C_{Ltot}\frac{V_{ofinal}}{\frac{1}{\beta}\cdot\frac{C_{Ltot}}{g_m}} > I_D \qquad \Rightarrow \frac{g_m}{I_D} > \frac{1}{\beta V_{ofinal}}$$

- Example: β=0.5, V_{ofinal}=0.5V → g_m/I_D > 4 S/A will result in slewing
- Very hard to avoid slewing, unless
  - We are willing to bias at very low g_m/I_D (power inefficient)
  - Feedback factor is small (large closed-loop gain, C_S/C_f)
  - Output voltage swing is small

Output Waveform with Initial Slewing

$$v_o(t) = \Delta V_{oslew} + \Delta V_{olin}\left(1 - e^{-(t-\Delta t_{slew})/\tau}\right)$$

$$v_o(t) = \frac{I_D}{C_{Ltot}}t = SR\cdot t$$



- Continuous derivative in the transition slewing→linear requires

$$\frac{I_D}{C_{Ltot}} = \frac{\Delta V_{olin}}{\tau} \qquad \Delta V_{olin} = \frac{\tau\cdot I_D}{C_{Ltot}}$$

Dynamic Error with Slewing

$$\Delta V_{oslew} = V_{ofinal} - \Delta V_{olin} \qquad \Delta t_{slew} = \left(V_{ofinal} - \Delta V_{olin}\right) \cdot \frac{C_{Ltot}}{I_D}$$

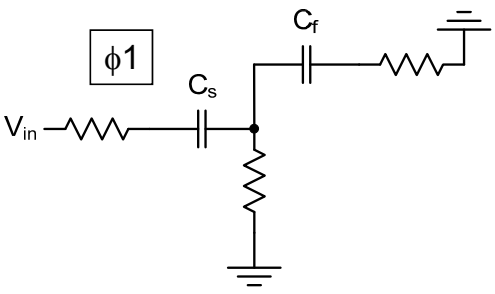
- Using the above result, we can now calculate the dynamic error during the final linear settling portion

$$\text{For } t > \Delta t_{slew}: \qquad v_o(t) = \Delta V_{oslew} + \Delta V_{olin}\left(1 - e^{-(t-\Delta t_{slew})/\tau}\right)$$

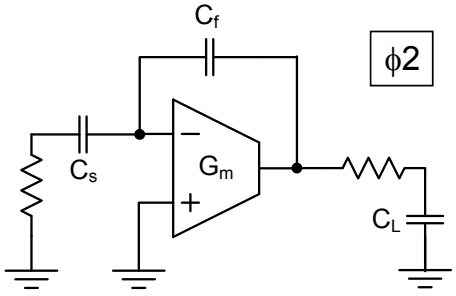
$$\varepsilon_d(t) = \frac{v_o(t) - V_{final}}{V_{final}} = \frac{\Delta V_{oslew} + \Delta V_{olin}\left(1 - e^{-(t-\Delta t_{slew})/\tau}\right) - V_{ofinal}}{V_{ofinal}}$$

$$\varepsilon_d(t) = -\frac{\Delta V_{olin}}{V_{ofinal}} e^{-(t-\Delta t_{slew})/\tau}$$

Noise Analysis

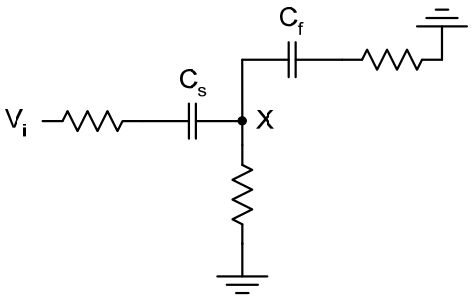


Noise due to switches



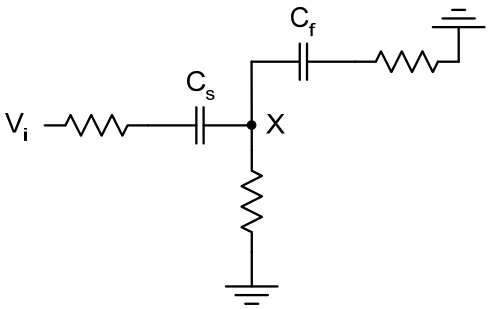
Noise due to amplifier and switches

Tracking Phase (ϕ1)



- Variable of interest is total integrated "noise charge" at node X, q_x^2
- Cumbersome to compute using standard analysis
  - Find transfer function from each noise source (3 resistors) to q_x
  - Integrate magnitude squared expressions from zero to infinity and add
- Much easier
  - Use equipartition theorem: each “degree of freedom” (e.g. energy stored on capacitor) of a system in thermal equilibrium holds an average energy of kT/2

Tracking Phase Noise Charge



- Energy stored at node X is

$$\frac{1}{2}\frac{q_x^2}{C_{eff}} = \frac{1}{2}\frac{q_x^2}{C_s + C_f}$$

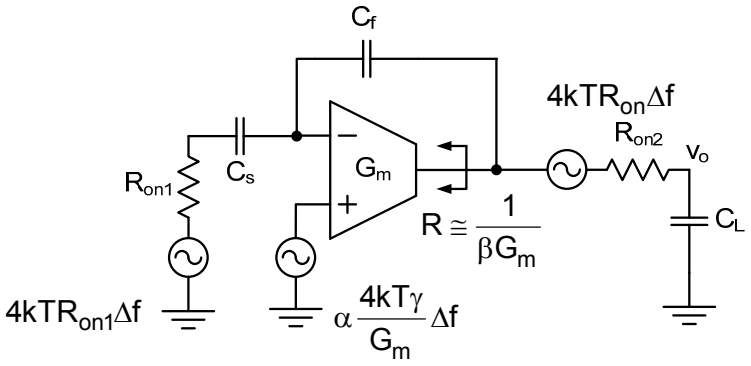
- Apply equipartition theorem

$$\overline{\frac{1}{2}\frac{q_x^2}{C_s + C_f}} = \frac{1}{2}kT$$

$$\overline{q_x^2} = kT\left(C_s + C_f\right)$$

- Note that any additional parasitic capacitance at node X will increase the sampled noise charge!

Redistribution Phase Noise



α>1 excess noise factor

- In a proper design, R_{on1} and R_{on2} will be much smaller than $1/\beta G_m$, else the switches would significantly affect the dynamics, which would be very wasteful
  - It is much easier to design switches with low on-resistance than an amplifier with very large G_m

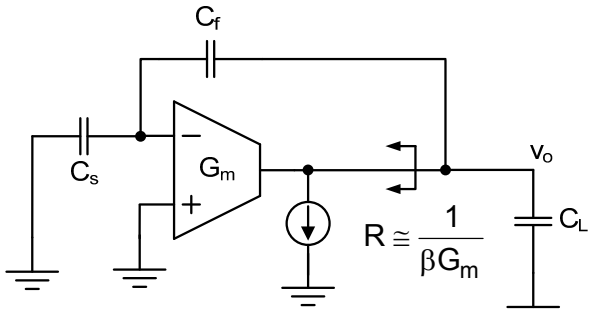
Output Referred Noise Comparison

- R_{on1} noise referred to v_o $$N_1 = 4kTR_{on1}\Delta f \cdot \left(\frac{C_s}{C_f}\right)^2 \cdot |H(j\omega)|^2$$
- R_{on2} noise referred to v_o $$N_2 = 4kTR_{on2}\Delta f \cdot |H(j\omega)|^2$$
- Amplifier noise referred to v_o $$N_a = \alpha \frac{4kT\gamma}{G_m}\Delta f \cdot \left(1+\frac{C_s}{C_f}\right)^2 \cdot |H(j\omega)|^2$$

$$\frac{N_a}{N_1} = \frac{\alpha\gamma}{G_m R_{on1}} \frac{\left(1+\frac{C_s}{C_f}\right)^2}{\left(\frac{C_s}{C_f}\right)^2} >> 1 \qquad \frac{N_a}{N_2} = \frac{\alpha\gamma}{G_m R_{on2}} \left(1+\frac{C_s}{C_f}\right)^2 >> 1$$

- Amplifier noise dominates over noise due to R_{on1}, R_{on2}

Total Integrated Amplifier Noise

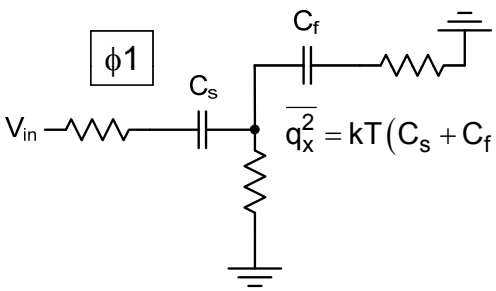


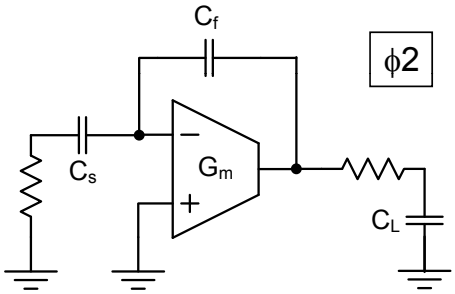
$$\alpha \cdot 4kT\gamma G_m \cdot \Delta f$$

$$\frac{\overline{v_o^2}}{\Delta f} = \alpha \cdot 4kT\gamma \frac{1}{\beta R} \cdot \left| R \| \frac{1}{j\omega C_{Ltot}} \right|^2$$

$$\overline{v_o^2} = \int_0^\infty \alpha \cdot 4kT\gamma \frac{1}{\beta R} \cdot \Delta f \cdot \left| \frac{R}{1 + j\omega RC_{Ltot}} \right|^2 df = \alpha\gamma \frac{1}{\beta} \frac{kT}{C_{Ltot}}$$

Adding up the Noise Contributions





$$\overline{v_{o,1}^2} = \frac{\overline{q_x^2}}{C_f^2} = kT\left(\frac{C_s + C_f}{C_f^2}\right) = \frac{kT}{C_f}\left(1 + \frac{C_s}{C_f}\right)$$

$$\overline{v_{o,2}^2} \cong \alpha\gamma \frac{1}{\beta} \frac{kT}{C_{Ltot}}$$

$$\boxed{\overline{v_{o,tot}^2} = \frac{kT}{C_f}\left(1 + \frac{C_s}{C_f}\right) + \alpha\gamma \frac{1}{\beta} \frac{kT}{C_{Ltot}}}$$

Noise in Differential Circuits

- In differential circuits, the noise power is doubled (because there are two half circuits contributing to the noise)
- But, the signal power increases by 4x
  - Looks like a 3dB win?

$$DR_{single} \propto \frac{\hat{V}_o^2}{\frac{kT}{C}} \qquad DR_{diff} \propto \frac{\left(2\hat{V}_o\right)^2}{2\frac{kT}{C}} = 2\frac{\hat{V}_o^2}{\frac{kT}{C}}$$

- Yes, there's a 3dB win in DR, but it comes at twice the power dissipation (due to two half circuits)
- Can get the same DR/power in a single ended circuit by doubling all cap sizes and g_m

Summary – Sampling Circuits

- Three predominant implementation styles
  - Purely passive
  - Source follower T/H, up to ~9-10bit accuracy
  - Charge redistribution or flip-around architecture
- In a typical, properly designed circuit only the most fundamental issues are significant
  - Jitter, kT/C noise
- Charge injection is not a problem if properly handled
  - E.g. through bottom plate sampling

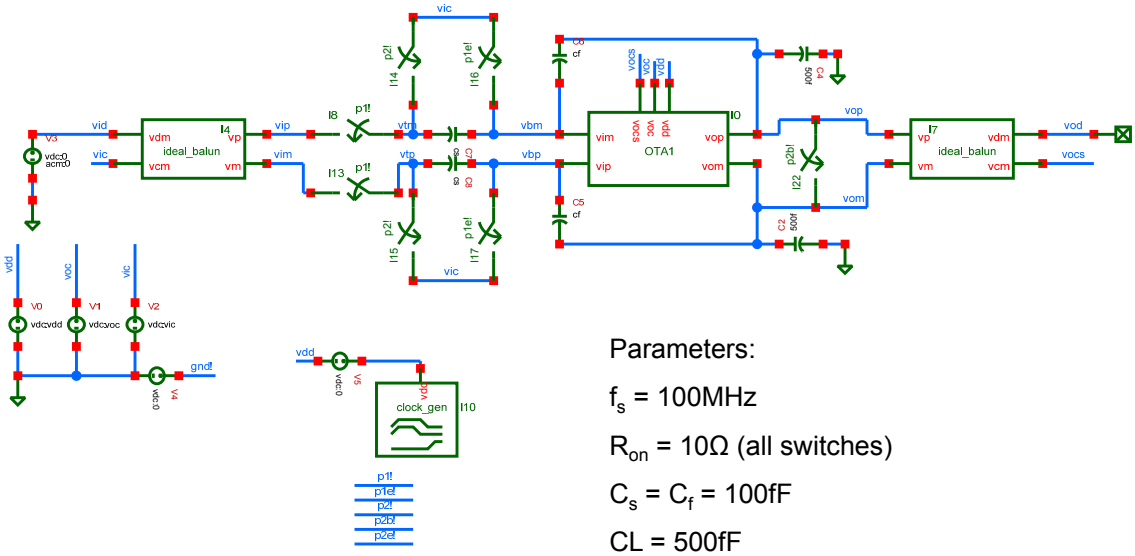
Outline

- Elementary track-and-hold circuit and its nonidealities
- First order improvements to elementary track-and-hold
- Advanced techniques
  - Clock bootstrapping
  - Bottom plate sampling
- Settling and noise analysis in charge-redistribution track-and-hold circuit
- Noise simulation example

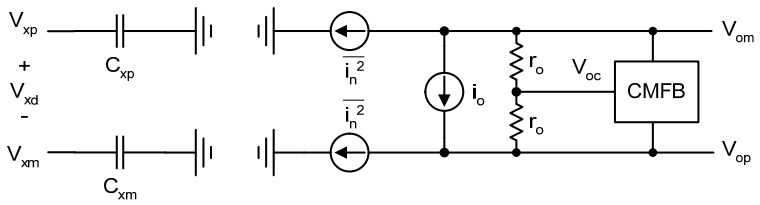
Noise Simulation Example

- Three ways to simulate noise in switched capacitor circuits
- Basic .ac/.noise Spice simulations
  - Must simulate noise in each clock phase separately
    - Activate $\phi 1$ switches, run .noise and integrate noise charge at relevant node over all frequencies and refer to output
    - Activate $\phi 2$ switches, run .noise and integrate noise at output
- Periodic Steady State Simulation
  - E.g. SpectreRF or BDA, "periodic noise analysis“ (PNOISE)
  - Allows to simulate noise while switched capacitor circuit is clocked between ϕ_1 and ϕ_2
    - Noise from all phases is automatically added, all correlation taken care of
- Transient Noise

Example Track and Hold Schematic



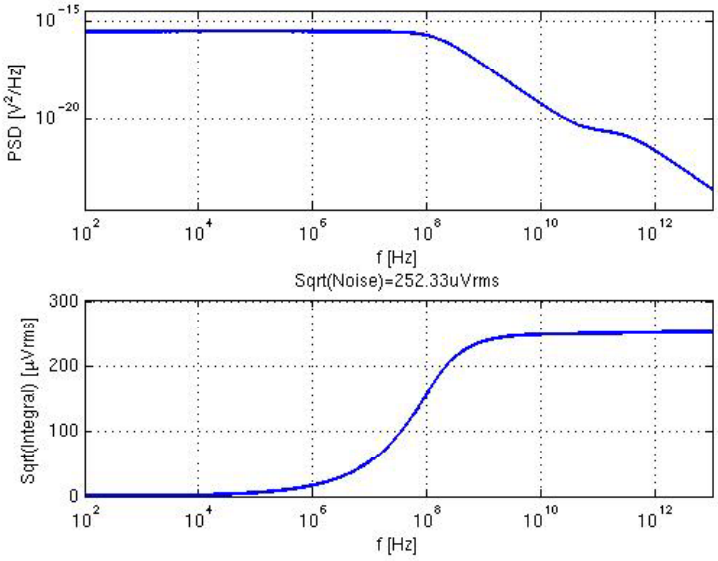
OTA Simulation Model



$$C_{xp,m} = \frac{g_m}{2\pi f_T} \qquad \frac{\overline{i_n^2}}{\Delta f} = \alpha \cdot kTg_m \qquad r_o = \frac{a_{vo}}{g_m} \qquad I_D = \frac{g_m}{(g_m / I_D)}$$

Parameters: g_m=1mS, a_{vo}=1000, α=2, g_m/I_D=10S/A , f_T=20GHz

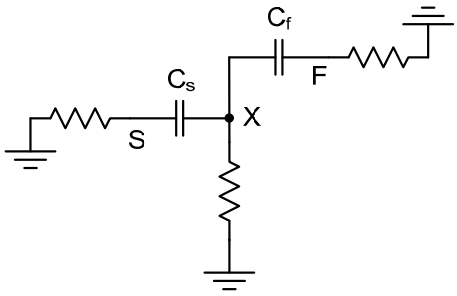
Hold Mode Noise Simulation (.noise)



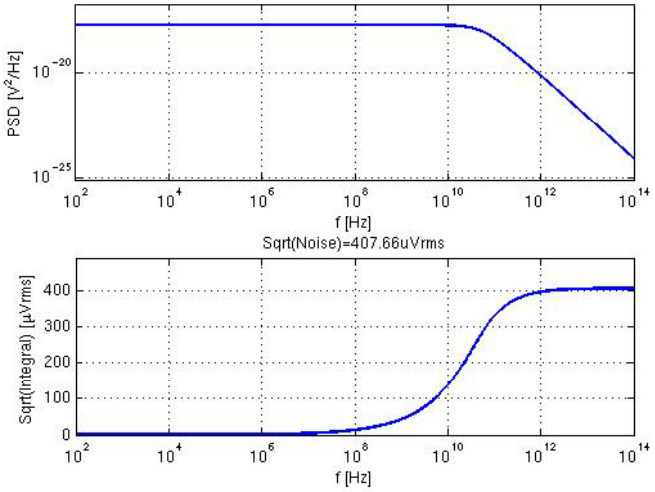
Calculated value: 248uVrms

Track Mode Noise Simulation (.noise)

```
*** Compute noise charge and refer charge referred to output via Cf
en vno 0  vcvs vol=( cs*v(x,s) + cf*v(x,f) )/cf'
.ac dec 100 100 100Gig
.noise v(vno) vdummy
```

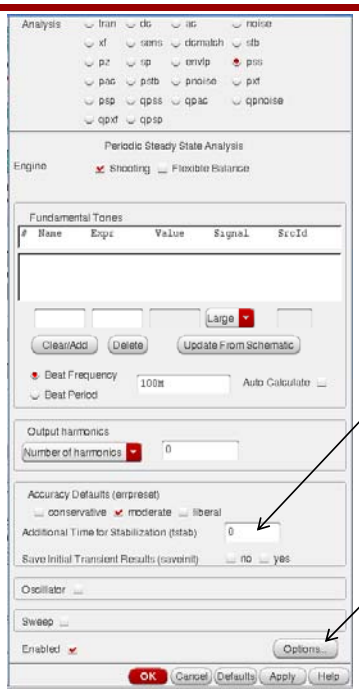


(shown single ended for simplicity)



Calculated value: 413uVrms

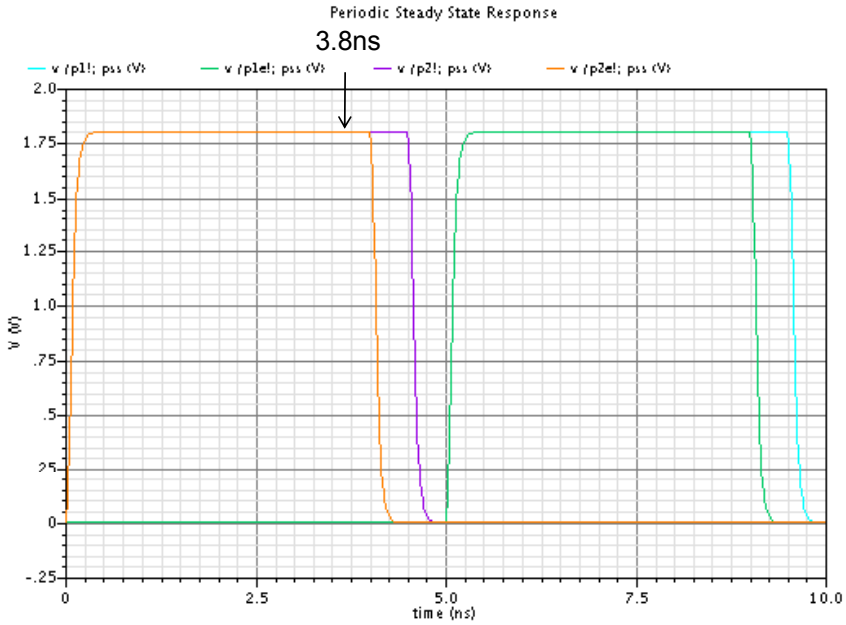
PSS Setup



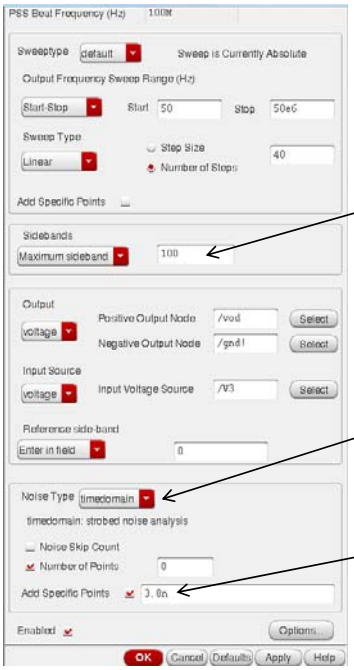
Use “tstab” if your circuit needs time to get into steady state (e.g. clock bootstrap circuits)

Important: set “maxacfreq” to the highest frequency at which you expect noise to be significant (10GHz in this example; see plot on previous slide!)

PSS Waveforms (Clocks)



PNOISE Setup



"Number of sidebands" – typically ~20…200 to handle noise folding properly. Fast switches → more sidebands needed. Again, Be sure to set "maxacfrequency" in the PSS analysis options to a correspondingly large value.

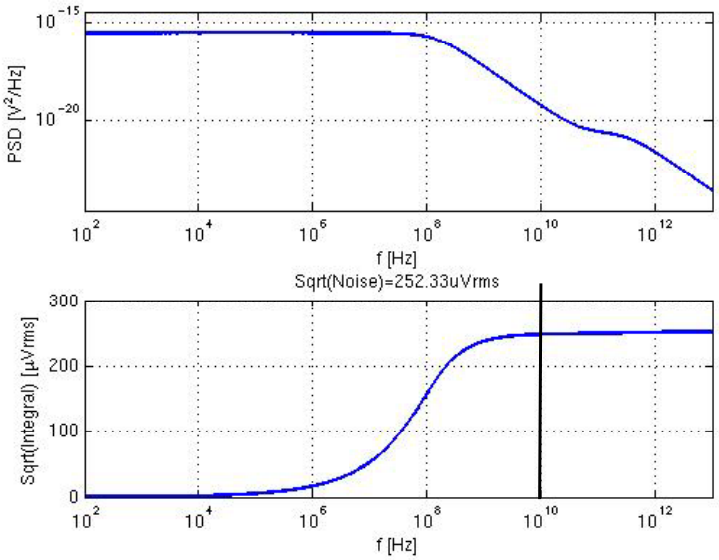
Note: This is not a problem in advanced simulators such as BDA, which cover an "infinite" number of sidebands

"timedomain" means simulator computes spectrum of discrete time noise samples

Sampling instant (3.8ns in this example

How Many Sidebands are Needed? (1)

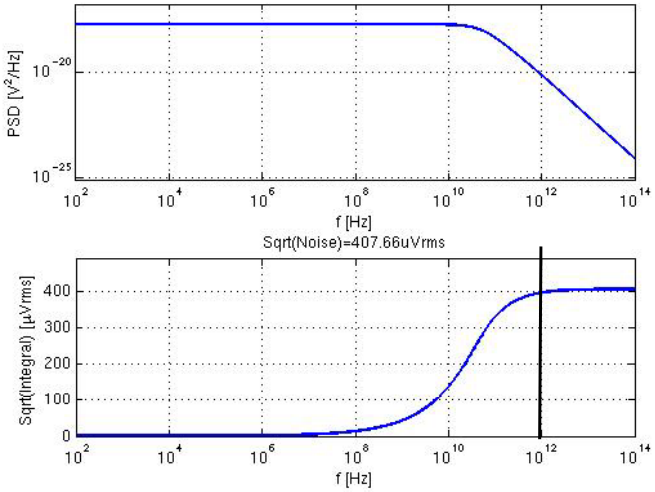
Hold mode noise integral (.noise)



Noise up to 10GHz must be considered !
→ numsidebands =100
→ macacfreq = 10GHz,

How Many Sidebands are Needed? (2)

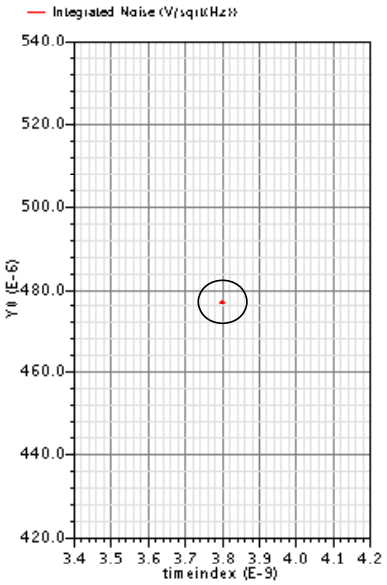
Track mode noise integral, (.noise, R_{on} = 10 Ohms)



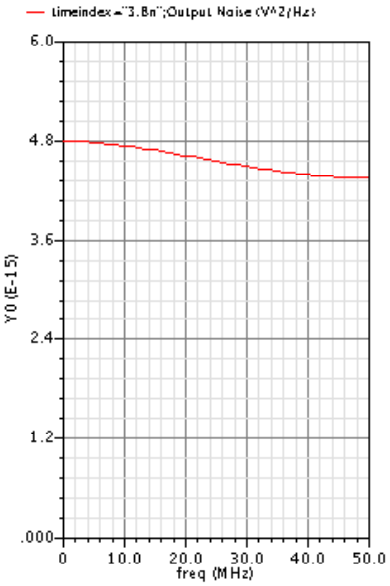
Noise up to 1 THz must be considered !
→ numsidebands =10,000?
No! Increase R_{on} to make maintain reasonable simulation time. Keep $R_{on}C$ ~ 10x faster than amplifier.

PNOISE Result

Sampled Noise



Noise PSD

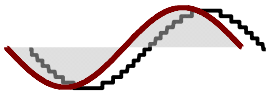


Comparison

Calculated:	$\overline{v_{o,1}^2} = (413\mu Vrms)^2$	$\overline{v_{o,2}^2} = (248\mu Vrms)^2$	$\overline{v_{o,tot}^2} = (482\mu Vrms)^2$
Simulated: (.noise)	$\overline{v_{o,1}^2} = (415\mu Vrms)^2$	$\overline{v_{o,2}^2} = (252\mu Vrms)^2$	$\overline{v_{o,tot}^2} = (485\mu Vrms)^2$
Simulated: (PNOISE)		$\overline{v_{o,tot}^2} = (478\mu Vrms)^2$	

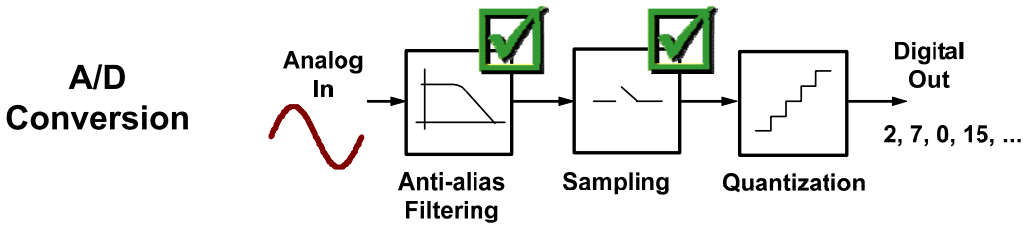
- Very good agreement between calculation and both simulation approaches

Voltage Comparators



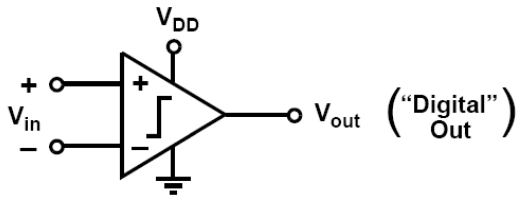
Katelijn Vleugels
Stanford University

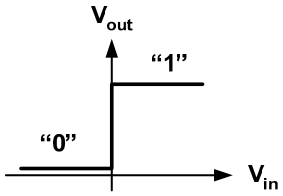
Recap



- Ultimately, building a quantizer requires circuit elements that "make decisions"
- The most widely used "decision circuit" is a voltage comparator

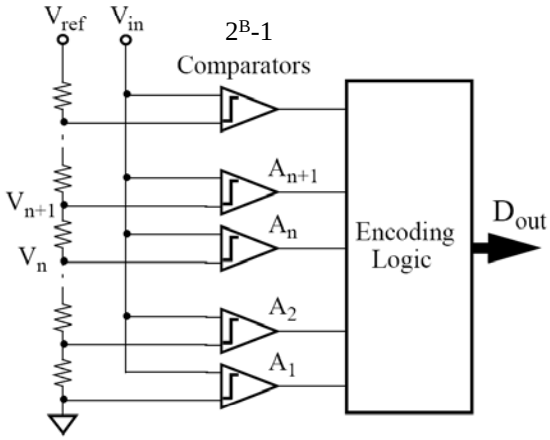
Ideal Voltage Comparator

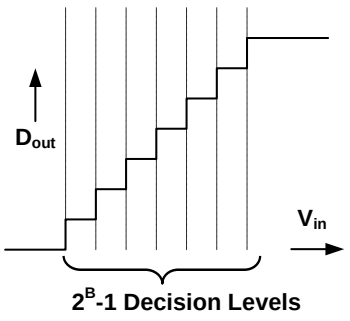




- Function
  - Compare the instantaneous values of two analog voltages (e.g. an input signal and a reference voltage) and generate a digital 1 or 0 indicating the polarity of that difference

Preview - Flash ADC

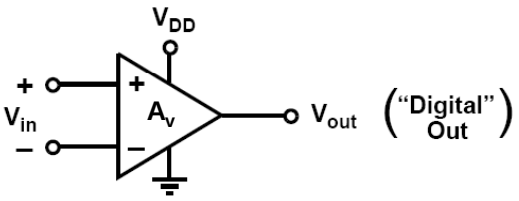


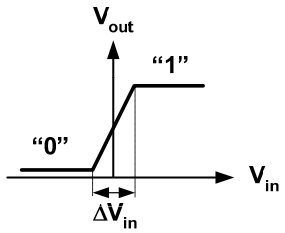


Design Considerations

- Accuracy
  - Gain (resolution)
  - Offset
- Speed
  - Small-signal bandwidth
  - Settling time or delay time, slew rate
  - Overdrive recovery
- Power dissipation
- Input properties
  - Sampled data versus continuous time
  - Common-mode rejection
  - Input capacitance and linearity of input capacitance
  - Kickback noise

Gain Requirements





$$\Delta V_{in} = \frac{V_{DD}}{A_{vo}}$$

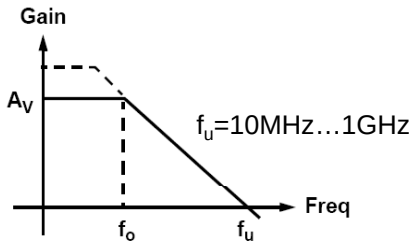
- E.g. 12-bit ADC, V_{DD}=1.8V, FSR=0.9V, $\Rightarrow$LSB=0.9V/4096
- For 1/2 LSB precision, we need

$$A_v = \frac{1.8V}{0.5 \cdot 0.9V / 4096} \cong 16{,}000 = 84dB$$

How to Implement High Gain?

- Considerations
  - Amplification need not be linear
  - Amplification need not be continuous in time, if comparator is used in a sampled data system
    - Clock signal will tell comparator when to make a decision
- Implementation options to be looked at
  - Single stage amplification
    - E.g. OTA or OpAmp in open loop configuration
  - Multi-stage amplification
    - E.g. cascade of resistively loaded differential pairs
  - Regenerative latch using positive feedback
    - E.g. cross coupled inverters

How about Using an OpAmp or OTA?

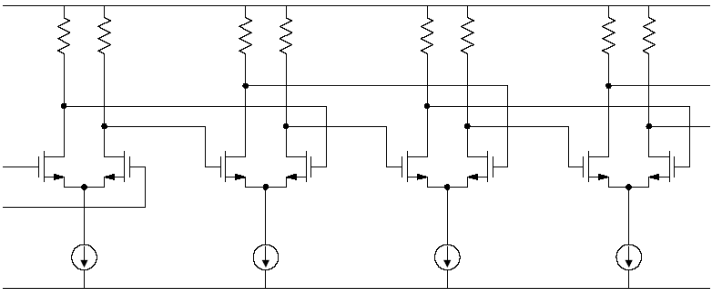


f_u = unity gain frequency, f_o = – 3dB frequency

$$f_o = \frac{f_u}{A_v} \approx \frac{1\text{GHz}}{16{,}000} = 62.5\text{kHz} \qquad \tau_o = \frac{1}{2\pi f_o} = 2.5\mu\text{s}$$

- Way too slow!

Cascade of Open-Loop Amplifiers



For each stage: $A_0 = g_m R$ $\omega_0 = \frac{1}{RC}$ $\omega_u = \frac{g_m}{C_{gs}} \cong \text{const.}$ $\boxed{\omega_0 = \frac{1}{RC} = \frac{\omega_u}{A_0}}$

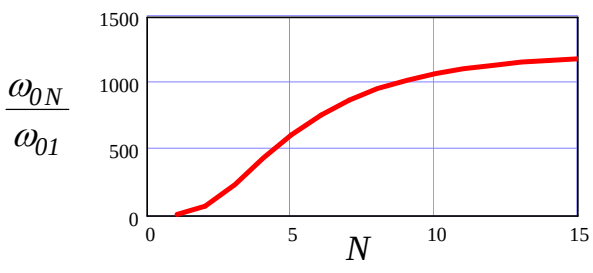
- Possible choices for a given, constant overall gain objective
  - Lots of stages with low gain
  - Only a few stages with moderate gain

Bandwidth Perspective

- If we only care about small signal bandwidth, it follows that we should cascade many low gain stages
  - Makes intuitive sense, because each individual stage will have a very large bandwidth
- Detailed analysis shows

$$\frac{\omega_{0N}}{\omega_{01}} = A_v^{\left(\frac{N-1}{N}\right)} \sqrt{2^{\frac{1}{N}} - 1}$$

A_v Total gain requirement
ω_{01} Bandwidth of single stage realization
ω_{0N} Bandwidth of N stage realization



(A_v=10,000)

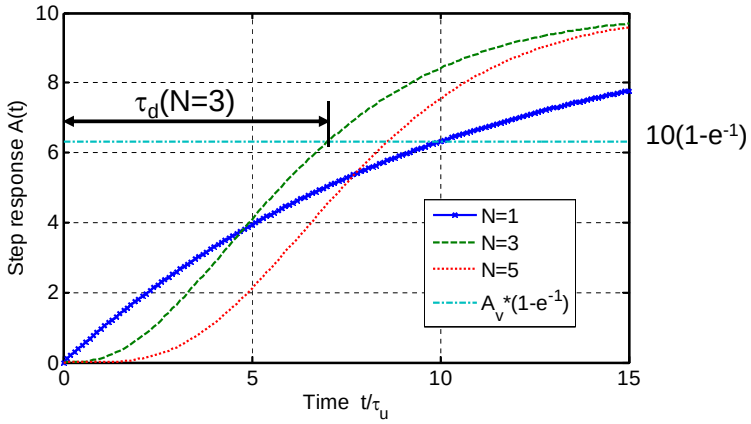
Step Response (1)

- When the input is a sampled data signal, it is more important to minimize the delay in response to an input step

$$V_{out}(s) = V_{in}(s)A(s) = \frac{V_{istep}}{s}\frac{A_v}{\left(1+s\cdot\tau_u A_v^{1/N}\right)^N} \qquad \tau_u = \frac{1}{\omega_u} \qquad A_v = A_0^N$$

$$V_{out}(t) = V_{istep}A_v\underbrace{\left(1-e^{-\frac{t}{\tau_u\cdot A_v^{1/N}}}\sum_{i=0}^{N-1}\frac{\left(\frac{t}{\tau_u\cdot A_v^{1/N}}\right)^i}{i!}\right)}_{A(t)}$$

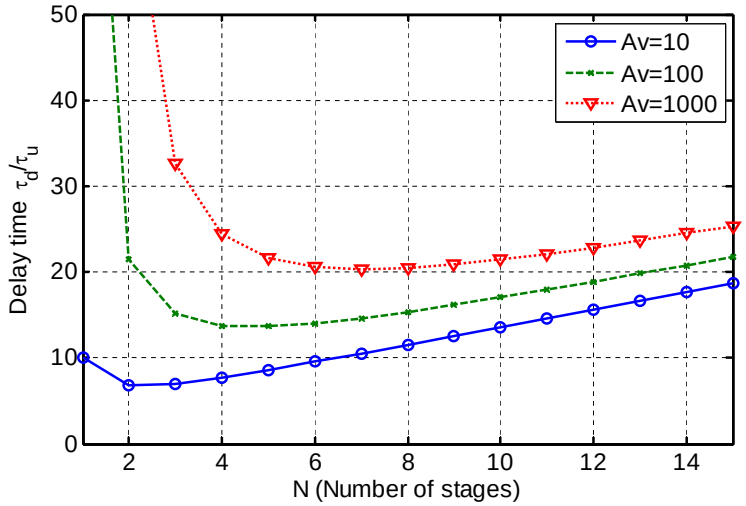
Step Response (2)



- τ_d = equivalent settling time constant of N stage realization
  - time required for N stage implementation to reach same output voltage as a single stage realization achieves after τ_0
- Three stage amplifier wins! (for A_v=10)

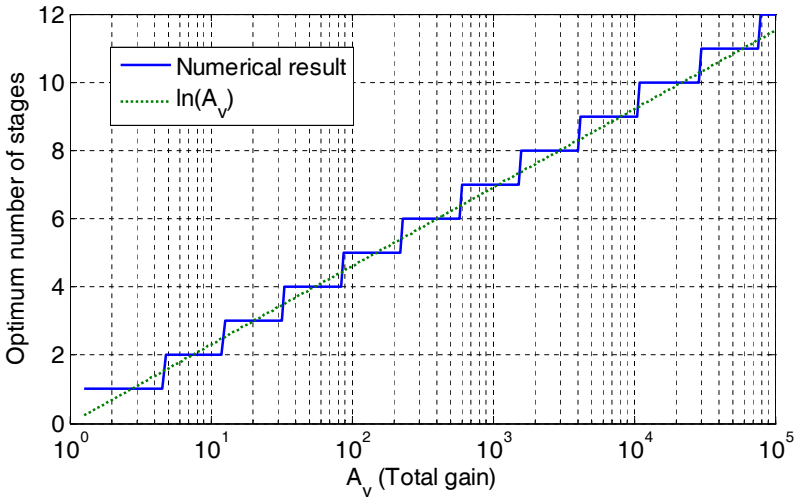
Delay versus Number of Stages

$$A(\tau_d) = A_v\left(1 - e^{-1}\right) \quad \Rightarrow \quad \tau_d \quad \text{(numerically)}$$



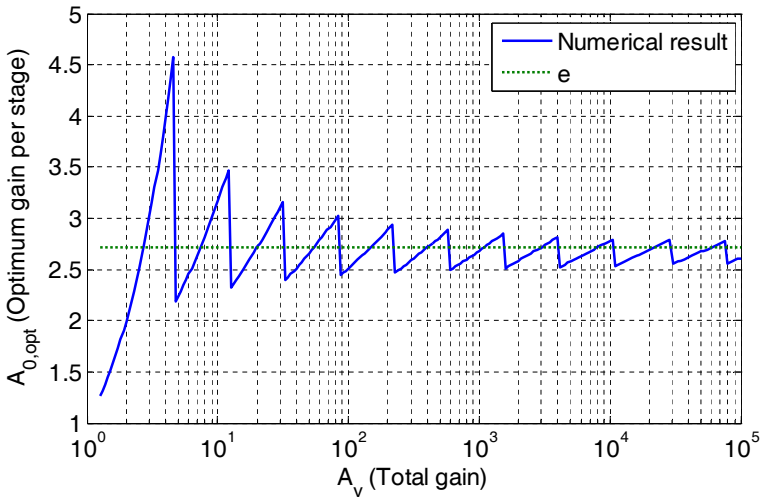
- Shallow minima!

Optimum Number of Stages



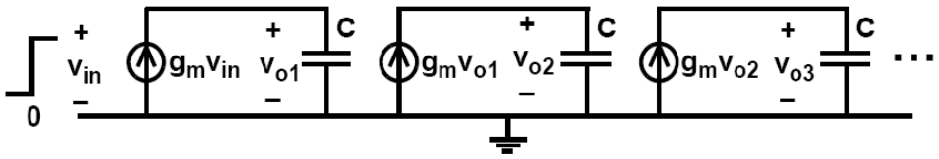
$$N_{opt} \cong \ln(A_v)$$

Optimum Gain per Stage



$$N_{opt} \cong \ln(A_v) \qquad e^{N_{opt}} \cong A_v = A_{0,opt}{}^{N_{opt}} \quad \Rightarrow \quad A_{0,opt} \cong e$$

Cascade of "Integrators" (1)



- Intuition
  - Load resistors (in cascade of open loop amplifiers) shunt current away from load capacitance; this slows down amplification
  - Drop assumption $A_v=A_0{}^N$ to see what happens…
- Analysis

$$V_{o1} = \frac{g_m}{sC} v_{in} = \frac{\omega_u}{s} v_{in} \qquad V_{oN} = \frac{\omega_u^N}{s^N} v_{in}$$

Cascade of “Integrators” (2)

$$v_{o1} = \frac{1}{C}\int_0^t g_m v_{in} dt = \frac{g_m}{C} v_{in} t$$

$$v_{o2} = \frac{1}{C}\int_0^t g_m v_{o1} dt = \frac{g_m}{C}\int_0^t \frac{g_m}{C} v_{in} dt = \frac{1}{2}\left(\frac{g_m}{C}\right)^2 v_{in} t^2$$

$$v_{o3} = \frac{1}{C}\int_0^t g_m v_{o2} dt = \frac{g_m}{C}\int_0^t \left[\frac{1}{2}\left(\frac{g_m}{C}\right)^2 v_{in} t^2\right] dt$$

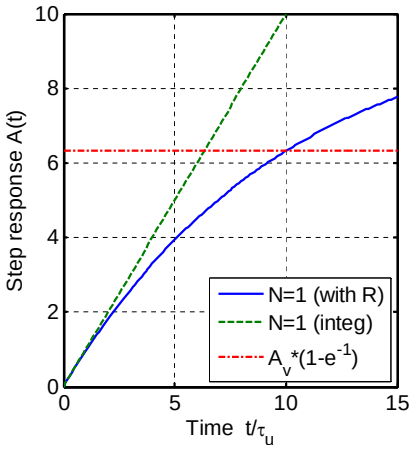
$$= \frac{1}{3}\left(\frac{1}{2}\right)\left(\frac{g_m}{C}\right)^3 v_{in} t^3$$

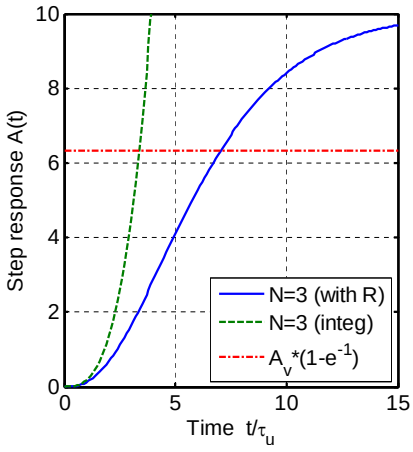
N Stages

$$v_{oN} = \left(\frac{g_m}{C}\right)^N \left(\frac{t^N}{N!}\right) v_{in}$$

Cascade of "Integrators" (3)

$$V_{out}(s) = \frac{V_{istep}}{s}\frac{\omega_u^N}{s^N} \qquad V_{out}(t) = V_{istep} \cdot \omega_u^N \frac{t^N}{N!}$$





Cascade of "Integrators" (4)

- Cascade of integrators achieves faster amplification than cascade of resistively loaded stages
- Delay time

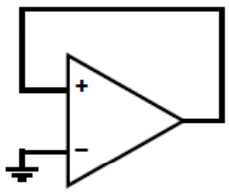
$$\tau_d = \tau_u \left[(N! \cdot A(\tau_d)) \right]^{1/N} \qquad A(\tau_d) = \frac{V_{out}(\tau_d)}{V_{instep}}$$

- Optimum number of stages approximately given by

$$N_{opt} = 1.1 \ln\left[A(\tau_d)\right] + 0.79$$ [Wu, JSSC 12/1988)

- Effective gain per stage is still relatively close to e=2.7183…

Latched Comparator (1)



$V_{out} = V_{in}$

$$g_m v = \frac{v}{R_L} + C\frac{dv}{dt}$$

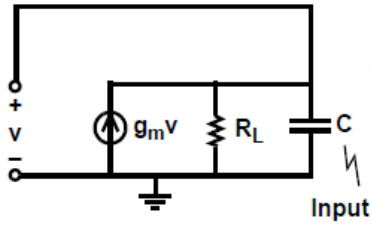
$$\frac{g_m}{C}\left(1 - \frac{1}{g_m R_L}\right) v = C\frac{dv}{dt}$$

$$\frac{g_m}{C}\left(1 - \frac{1}{g_m R_L}\right)\int_{t_1}^{t_2} dt = \int_{v_1}^{v_2} \frac{1}{v} dv$$

Latch Delay

$$\tau_D = t_2 - t_1 = \frac{C}{g_m}\left(\frac{1}{1 - \frac{1}{g_m R_L}}\right) \ln\left(\frac{v_2}{v_1}\right)$$

Ac Model:



$v_{out} = v_{in} \equiv v$

Input + Output C

where $\frac{v_2}{v_1} \equiv A_L$ = "LATCH GAIN"

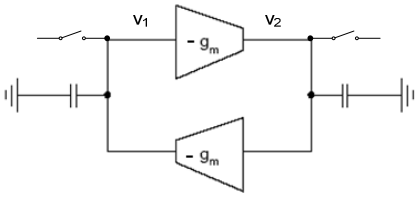
For $g_m R_L >> 1$

$$\tau_D \cong \left(\frac{C}{g_m}\right) \ln(A_L)$$

Latched Comparator (2)

$t < 0$ setup initial condition $v_{10} - v_{20} = v_{d0}$

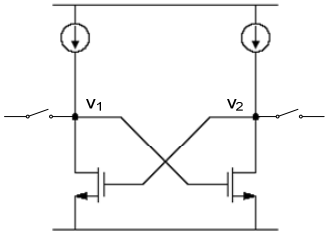
$t \geq 0$ enable positive feedback



$$\frac{dv_1}{dt} = \frac{i_1(t)}{C} = \frac{-g_m v_2(t)}{C}$$

$$\frac{dv_2}{dt} = \frac{i_2(t)}{C} = \frac{-g_m v_1(t)}{C}$$

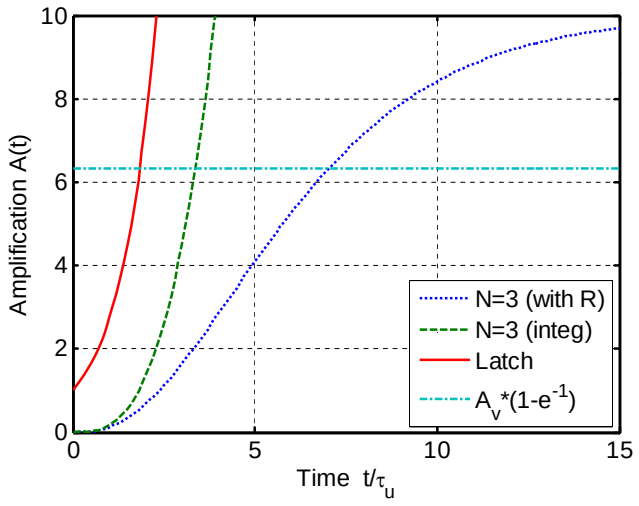
$$\tau_u = \frac{C}{g_m}$$



$$\Rightarrow v_1(t) - v_2(t) = v_d(t) = v_{d0} \cdot e^{t/\tau_u}$$

$$\Rightarrow A(t) = \frac{v_d(t)}{v_{d0}} = e^{t/\tau_u}$$

Comparison

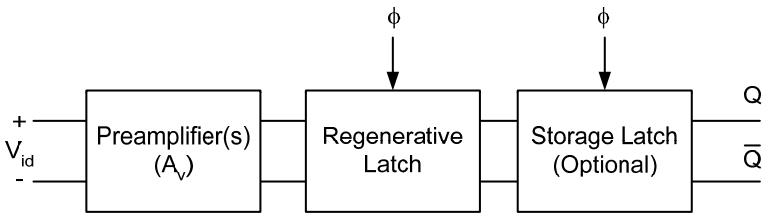


- Latch is much faster than cascade of amplifiers/integrators

Latch "Gain"

$A(\tau_d)$	τ_d/τ_u
10	2.3
100	4.6
1,000	6.9
10,000	9.2

"The" Architecture



- Why bother using pre-amplification (A_v)?
  - Offset
    - Hard to build latches with offset < 10…100mV
    - Use pre-amplification to lower input referred offset
  - Common mode rejection
  - Attenuate "kickback noise"
  - Metastability

Metastability (1)

- References
  - Veendrick, JSSC 4/1980
  - Zojer, JSSC 6/1985
- Consider minimum initial latch input voltage needed to regenerate to V_{DD} within maximum available time T_{max}

$$V_{d0\,min} = \frac{V_{DD}}{e^{T_{max}/\tau_u}}$$

- Minimum required pre-amplifier input

$$V_{id0\,min} = \frac{1}{A_v}\frac{V_{DD}}{e^{T_{max}/\tau_u}}$$

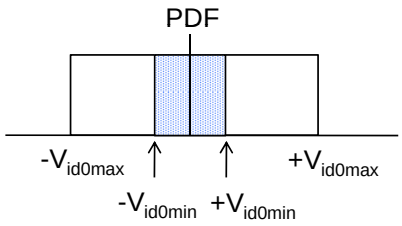
- Probability of seeing a metastable output

$$P(\text{Error}) = P\left(V_{id0} < V_{id0\,min}\right)$$

Metastability (2)

- Assuming a uniform input signal distribution over some range, we have

$$P(\text{Error}) = \frac{V_{id0\,min}}{V_{id0\,max}}$$



- In a flash ADC, only one out of all comparators can be metastable. The effective input range V_{id0max} over which the signal is distributed is therefore ½ LSB ($\Delta/2$)

$$\therefore P(\text{Error}) = \frac{V_{id0\,min}}{V_{id0\,max}} = \frac{\frac{1}{A_v}\frac{V_{DD}}{e^{T_{max}/\tau_u}}}{\frac{\Delta}{2}} = \frac{\frac{1}{A_v}\frac{V_{DD}}{e^{T_{max}/\tau_u}}}{\frac{V_{FS}}{2\cdot 2^B}} = \frac{2^{B+1}}{A_v}\frac{V_{DD}}{V_{FS}}e^{-T_{max}/\tau_u}$$

Metastability (3)

- Example: 6-bit, 500MHz Flash ADC, $T_{max}=T_s/2=1ns$, $\tau_u=1/(2\pi\cdot 5GHz)=32ps$, $A_v=3$, $V_{FS}=0.5V_{DD}$

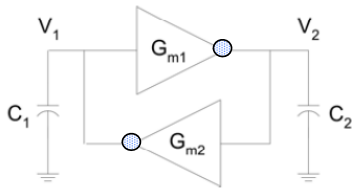
$$P(\text{Error}) = \frac{2^{6+1}}{3}\cdot 2\cdot e^{-1000/32} \cong 2\cdot 10^{-12}$$

- Mean time to failure (MTF)

$$MTF = \frac{1}{P(\text{Error})\cdot f_s} = \frac{1}{2\cdot 10^{-12}\cdot 0.5\cdot 10^{9}}\,s = 1000s \cong 16 \text{ minutes}$$

- Ideally design for MTF > 1…10 years (not always possible)
- Can improve MTF by
  - Reducing speed (larger T_{max}/τ_u)
    - Exponential dependence
  - Adding pre-amplifier gain
    - Linear dependence

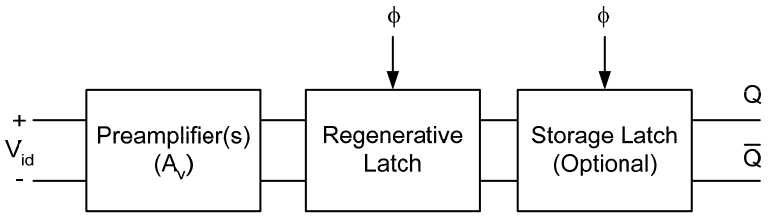
Latch: Dynamic Offset



$$\Delta C = C_1 - C_2$$

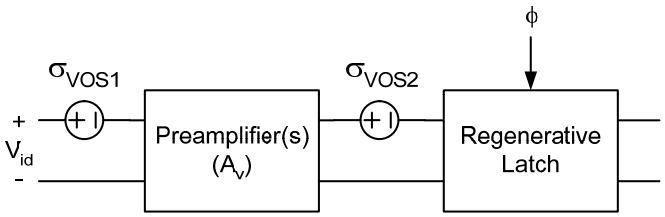
- $C_1 \neq C_2$ causes dynamic offset
- Can show $V_{os} \cong 0.5\cdot\Delta C/C\cdot(V_{(t=0)}-V_S)$; V_S = inverter switching voltage
  - Nikoozadeh & Murmann, IEEE TCAS II, Dec. 2006.
- Example
  - 0.5·10fF/100fF·(1V-0.5V)= 25mV (!)

CMOS comparators



- Dominant practical issues in comparator design are often
  - Offset
    - Latch offset often means must use one or more stages of low gain pre-amplification
    - Preamplifier offset often means must use offset cancellation techniques
  - Overdrive recovery
    - Time to recover from large signal and correctly detect small signal
    - Typically limits achievable comparator speed

Input Referred Offset



$$\sigma_{VOS}^2 = \sigma_{VOS1}^2 + \frac{1}{A_v^2}\sigma_{VOS2}^2$$

- Example: σ_{VOS1}=3mV, σ_{VOS2}=30mV, A_v=10

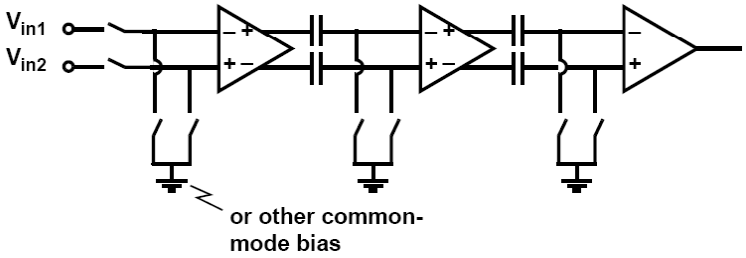
$$\sigma_{VOS} = \sqrt{(3\text{mV})^2 + \frac{1}{10^2}(30\text{mV})^2} = 4.2\text{mV}$$

Amplifier Offset Cancellation (1)

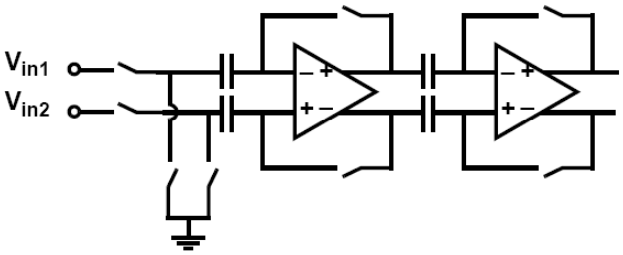
- In a sampled-data cascade of MOS amplifiers, offset voltages can be cancelled by storing them on ac-coupling capacitors in series with the amplifier stages
- The offset associated with a specific amplifier stage can be cancelled by storing it in series with either the input or the output of the stage
  - Output Series Cancellation:
    - short the amplifier inputs
    - store amplified offset on output coupling capacitors
  - Input Series Cancellation:
    - close a unity-gain loop around the amplifier
    - store offset on input coupling capacitors

Amplifier Offset Cancellation (2)

OUTPUT SERIES CANCELLATION

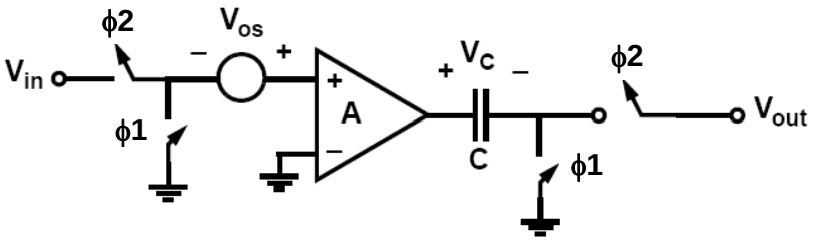


INPUT SERIES CANCELLATION



Output Series Cancellation

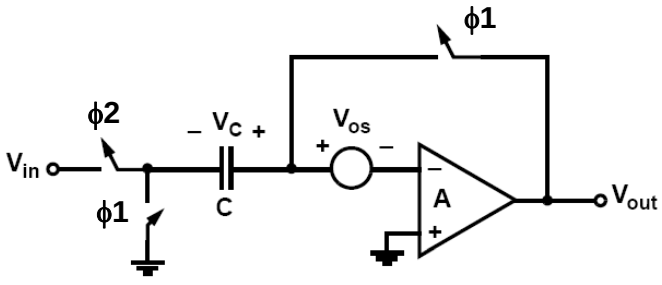
Ref: Poujois, et al., JSSC 8/78



- Phase 1: Offset storage, phase 2: Amplify
- Design considerations
  - Must ensure that amplifier does not saturate during phase 1
  - Must make C sufficiently large to avoid attenuation and mitigate charge injection error

Input Series Cancellation

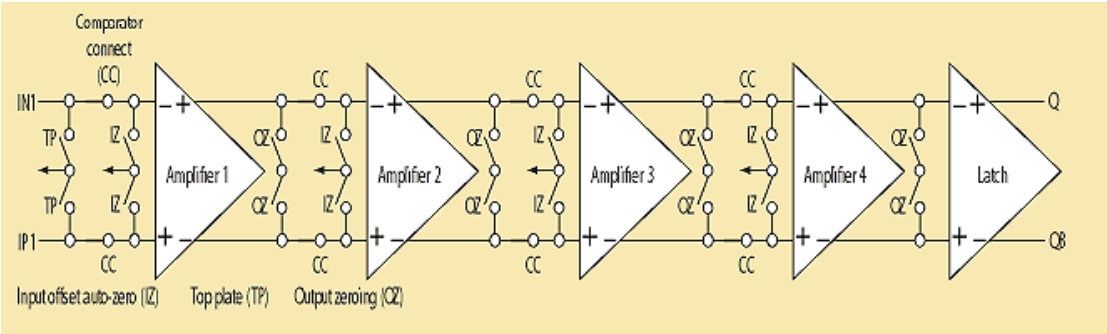
Refs: McCreary & Gray
Yee, et al.



- Phase 1: Offset storage, phase 2: Amplify
- In phase 2, input referred offset is $\approx V_{os}/(A+1)$
  - 4x reduction if A=3

Commercial Example: AD7671

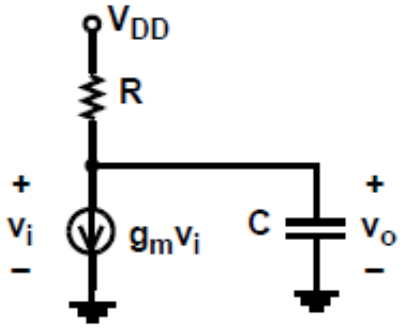
[http://www.elecdesign.com/Articles/Index.cfm?ArticleID=3956]



- Used in 16-bit, 1 MS/s successive approximation ADC, 0.6 μm CMOS technology
- Uses cascaded output series offset cancellation
- Offset <3 mV (over process, temperature)

Overdrive Recovery (1)

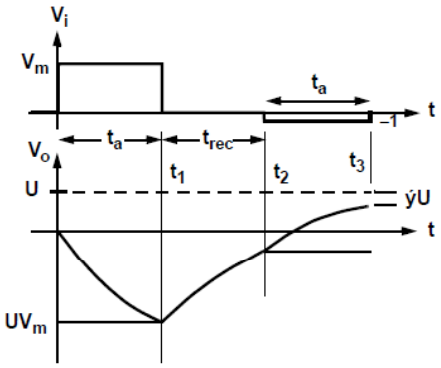
- Simple linear model of a single-pole amplifier



- During “reset”, the amplifier settles exponentially to its zero-input condition with a time constant $\tau = RC$.

Overdrive Recovery (2)

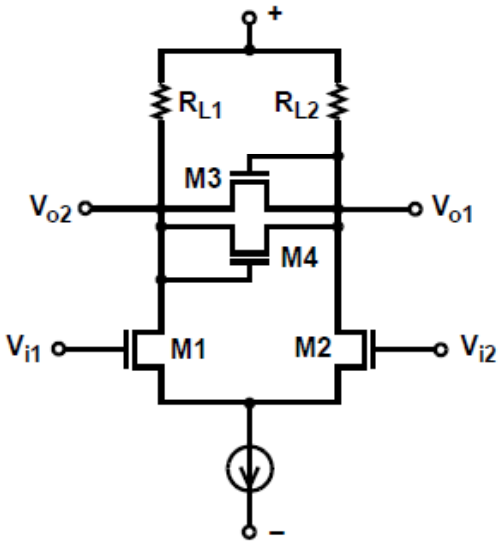
- Worst case scenario: maximum input V_m (normalized to ½ LSB) is followed by minimum input (½ LSB)



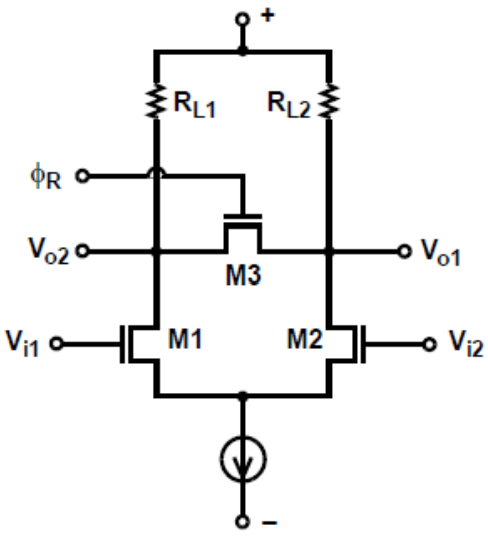
- To speed up overdrive recovery
  - Use low gain per stage
  - Use passive clamps
  - Use active restore

Overdrive Recovery (3)

Passive Clamps

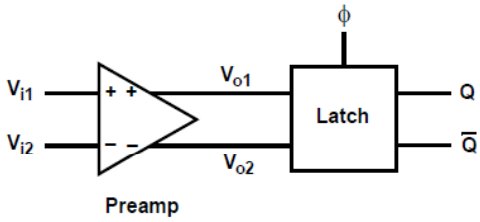


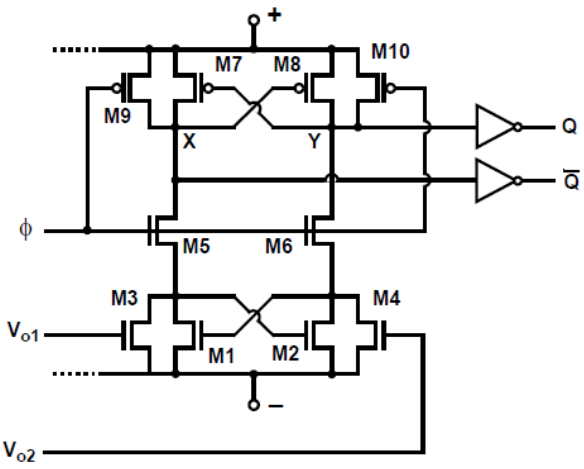
Active Restore



Comparator Examples (1)

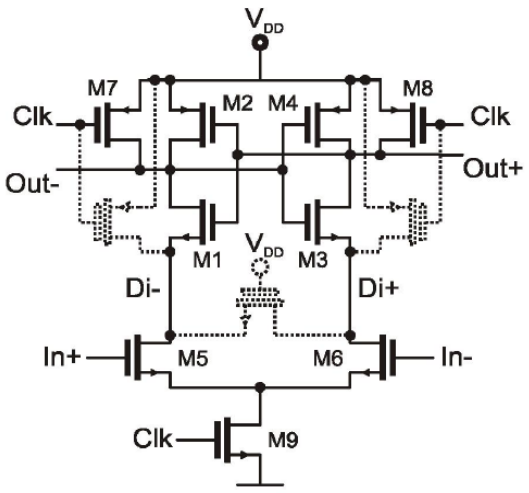
- Yukawa, JSSC 6/1985
  - Preamplifier followed by interesting latch
  - No offset cancellation
  - 8 bits, fs = 15MHz





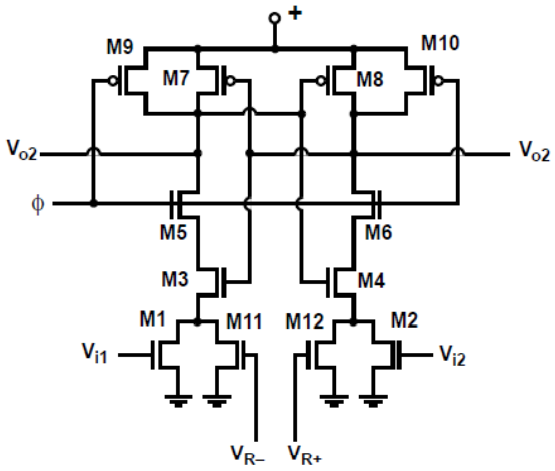
Comparator Examples (2)

- Purely dynamic "sense amplifier"
  - No DC current



Comparator Examples (3)

- Cho, JSSC 3/1995
  - Variation of Yukawa latch used as a dynamic comparator
  - No DC power dissipation when ϕ is high



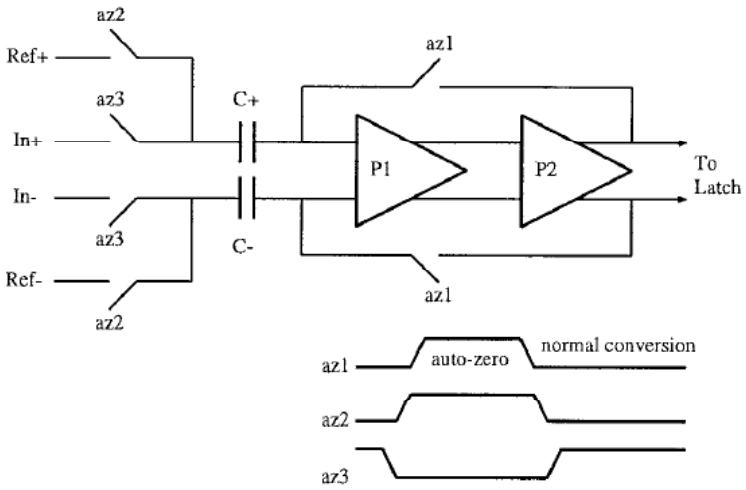
- M1, M2, M11 and M12 in linear (triode) region
- M11, M12 set differential threshold of comparator

$$V_{th} = \left(\frac{W_{11}}{W_1}\right) V_{ref}$$

where $V_{in} = V_{i1}-V_{i2}$ and $V_{ref} = V_{R+}-V_{R-}$

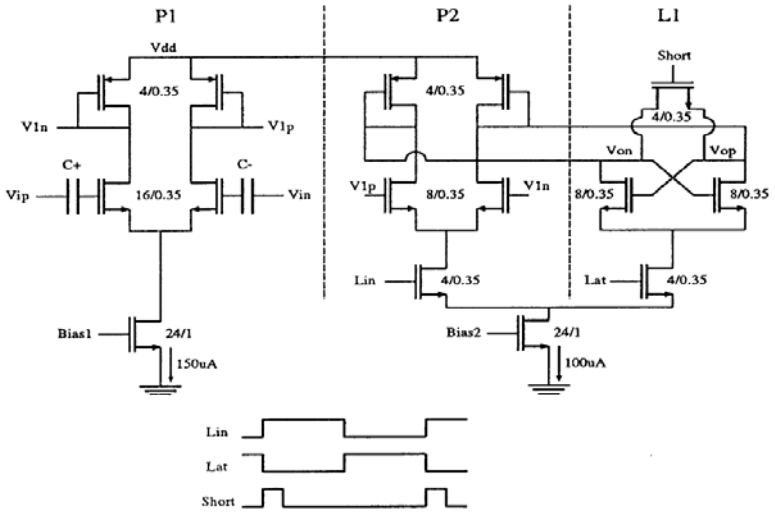
Comparator Examples (4)

- Mehr & Dalton, JSSC 7/1999



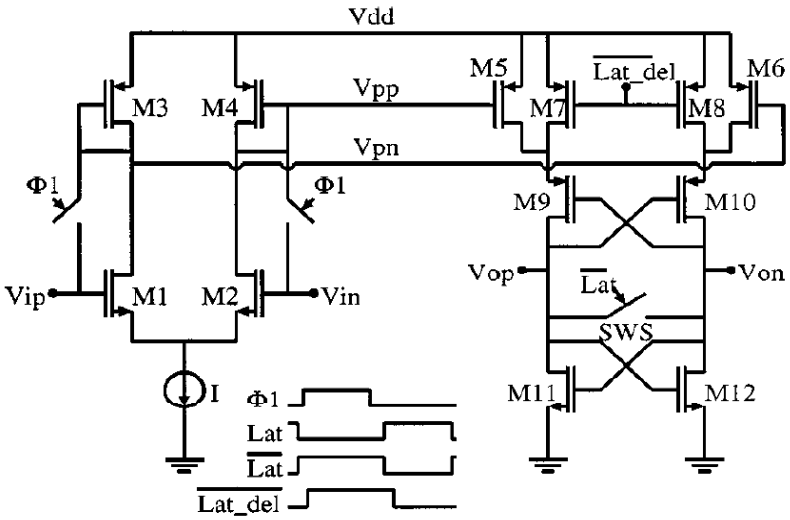
Comparator Examples (5)

- Mehr & Dalton, JSSC 7/1999



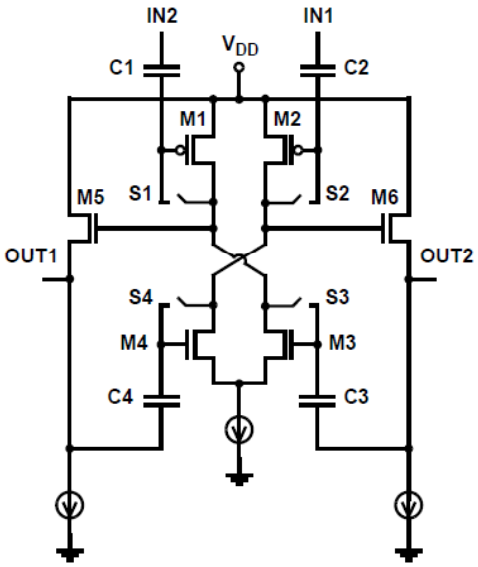
Comparator Examples (6)

- Mehr & Singer, JSSC 3/2000



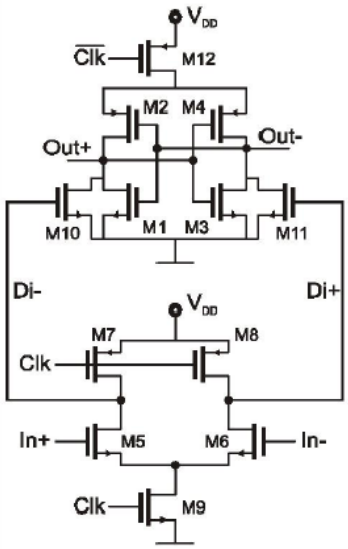
Comparator Examples (7)

- Wu, JSSC 12/1988
  - Instead of pre-amplification, uses latch with offset compensation



Comparator Examples (8)

- Schinkel, ISSCC 2007: "Double tail sense amplifier"



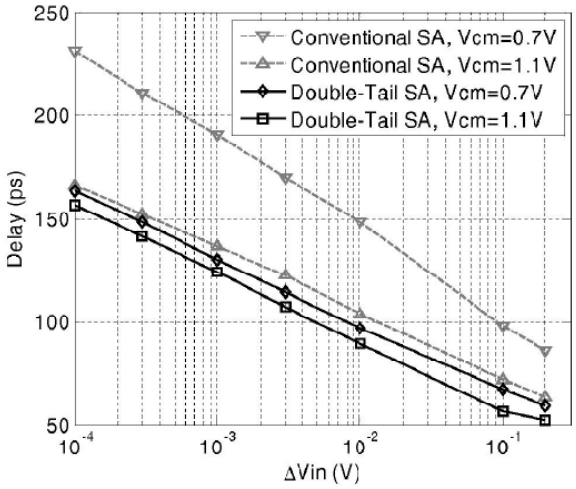


Figure 17.7.3: Simulated sense amplifier delays versus differential input voltage. The delay is the time between the clock edge and the instant when ΔOut crosses 1/2 V_{DD}.

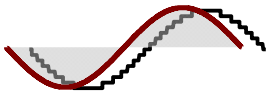
Selected References (1)

1. R. Poujois and J. Borel, “A Low Drift Fully Integrated MOSFET Operational Amplifier,” *IEEE J. of Solid-State Circuits*, pp. 499-503, Aug. 1978.
2. H.-S. Lee, D. A. Hodges and P. R. Gray, “A Self-Calibrating 15 Bit CMOS A/D Converter,” *IEEE J. of Solid-State Circuits*, vol. SC-19, pp. 813-819, Dec. 1984.
3. J. L. McCreary and P. R. Gray, “All-MOS Charge Redistribution Analog-to-Digital Conversion Techniques — Part I,” *IEEE J. of Solid-State Circuits*, vol. SC-10, no. 6, pp. 371-379, Dec. 1975.
4. Y. S. Yee, L. M. Terman and L. G. Heller, “A 1mV MOS Comparator,” *IEEE J. of Solid-State Circuits*, vol. SC-13, pp. 294-297, June 1978.
5. A. Yukawa, “A CMOS 8-Bit High-Speed A/D Converter IC,” *IEEE J. of Solid-State Circuits*, vol. SC-20, pp. 775-779, June 1985.
6. B. J. McCarroll, C. G. Sodini, and H.-S. Lee, “A High-Speed CMOS Comparator for Use in an ADC,” *IEEE J. of Solid-State Circuits*, vol. 23, pp. 159-165, Feb. 1988.
7. J.-T. Wu and B. A. Wooley, “A 100-MHz Pipelined CMOS Comparator,” *IEEE J. Solid-State Circuits*, vol. 23, pp. 1379-1385, Dec. 1988.
8. B. Razavi and B. A. Wooley, “A 12-b 5-MSample/s Two-Step CMOS A/D Converter,” *IEEE J. Solid-State Circuits*, vol. 27, pp. 1667-1678, Dec. 1992.
9. B. Razavi and B. A. Wooley, “Design Techniques for High-Speed, High-Resolution Comparators,” *IEEE J. Solid-State Circuits*, vol. 27, pp. 1916-1926, Dec. 1992.
10. T.B. Cho and P.R. Gray, “A 10-b, 20-Msample/s, 35-mW Pipeline A/D Converter,” IEEE J. Solid-State Circuits, vol. 30, pp. 166-172, Mar. 1995.

Selected References (2)

11. M. Choi and A. A. Abidi, "A 6-b 1.3-GSample/s A/D converter in 0.35-µm CMOS," *IEEE J. Solid-State Circuits,* pp. 1847-1858, Dec. 2001.
12. I. Mehr and D. Dalton, "A 500-MSample/s, 6-bit Nyquist-rate ADC for disk-drive read-channel applications," *IEEE J. Solid-State Circuits*, pp. 912-920, July 1999.
13. I. Mehr and L. Singer, “A 55-mW, 10-bit, 40-MSample/s Nyquist-Rate CMOS ADC,” IEEE J. Solid-State Circuits, pp. 318-25, March 2000.
14. H. J. M. Veendrick, “The Behavior of Flip-Flops Used as Synchronizers and Prediction of Their Failure Rate,” *IEEE J. Solid-State Circuits*, April 1980.
15. B. Zojer, et al., “A 6-bit/200-MHz full Nyquist A/D converter,” *IEEE J. Solid-State Circuits*, vol. SC-20, pp. 780-786, June 1985.
16. K.-L.J. Wong and C.-K.K. Yang, "Offset compensation in comparators with minimum input-referred supply noise," *IEEE J. Solid-State Circuits*, vol.39, pp. 837-840, May 2004.
17. A. Graupner, "A Methodology for the Offset-Simulation of Comparators," http://www.designers-guide.org/Analysis/comparator.pdf.
18. D. Schinkel et al., "A Double-Tail Latch-Type Voltage Sense Amplifier with 18ps Setup+Hold Time," ISSCC Dig. Techn. Papers, pp. 314-315, 2007.
19. P.P. Nuzzo, et al., "Noise Analysis of Regenerative Comparators for Reconfigurable ADC Architectures, IEEE Trans. Circuits Syst. I, pp.1441-1454, July 2008.
20. B.S. Leibowitz, et al., "Characterization of Random Decision Errors in Clocked Comparators," Proc. IEEE CICC, pp.691-694, Sep. 2008.

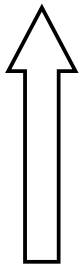
Flash and Folding & Interpolating ADCs



Katelijn Vleugels
Stanford University

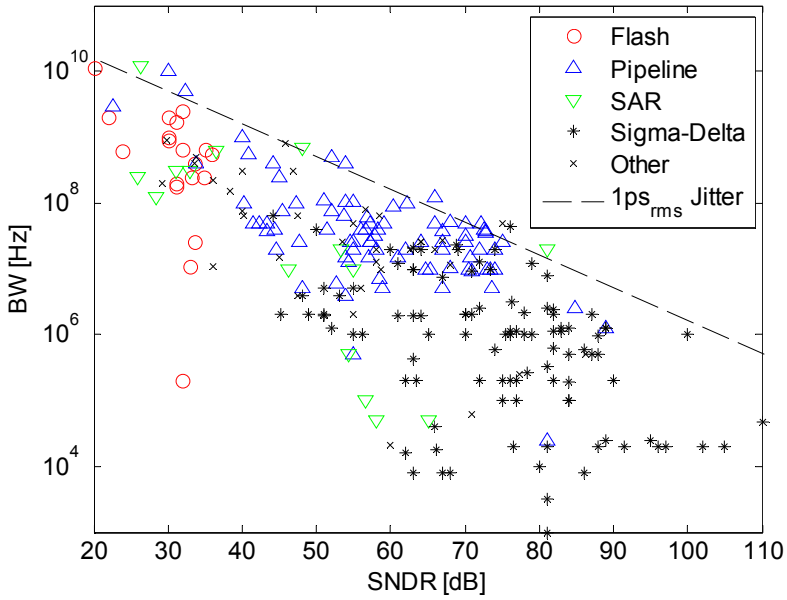
Nyquist ADC Architectures

- Nyquist rate
  - Word-at-a-time
    - **Flash ADC**
    - Instantaneous comparison with 2^B-1 reference levels
  - Multi-step
    - E.g. pipeline ADCs
    - Coarse conversion, followed by fine conversion of residuum
  - Bit-at-a-time
    - E.g. successive approximation ADCs
    - Conversion via a binary search algorithm
  - Level-at-a-time
    - E.g. single or dual slope ADCs
    - Input is converted by measuring the time it takes to charge/discharge a capacitor from/to input voltage

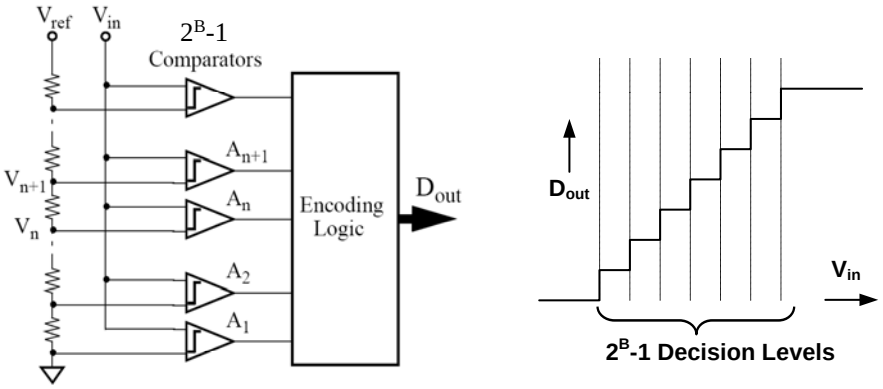


ADC Performance Survey (ISSCC & VLSI 97-08)

Data: http://www.stanford.edu/~murmann/adcsurvey.html



Flash ADC

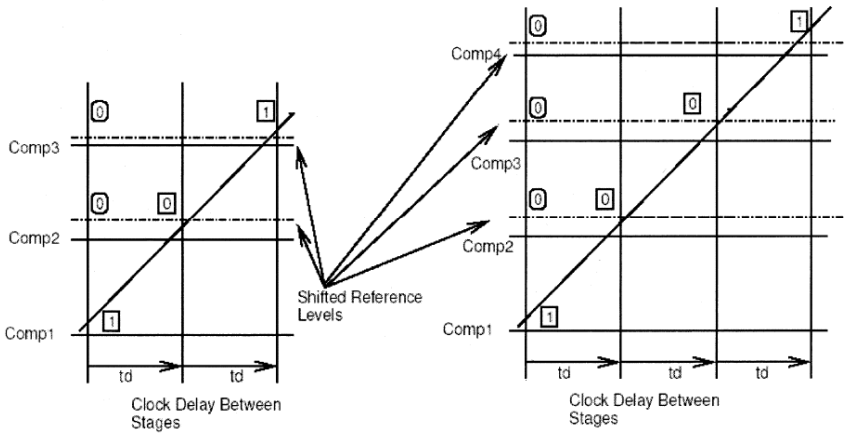


- Fast
  - Speed limited by single comparator plus encoding logic
- Exponential dependence of power, area and input capacitance upon number of bits
  - Typically use for resolution up to 6 bits

Limiting Error Sources

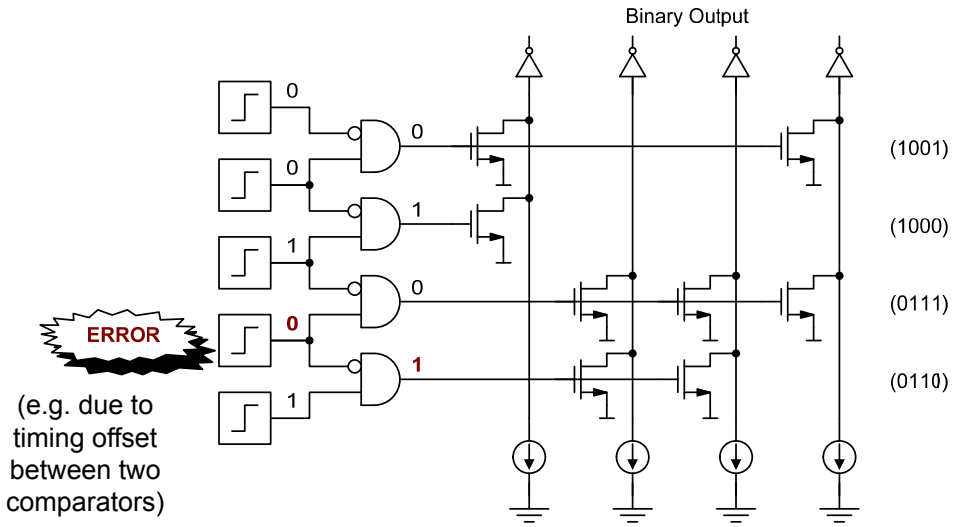
- Comparator input
  - Offset
  - Nonlinear input capacitance
  - Kickback noise (disturbs reference)
    - All comparators switching simultaneously
- Timing and comparator output
  - "Sparkle codes" caused by bubbles (… 111101000 …)
    - E.g. due to clock timing skew
  - Metastability
    - Analog Devices application note: "Find Those Elusive ADC Sparkle Codes and Metastable States" http://www.analog.com/en/content/0,2886,760%255F788%255F91218,00.html

Bubbles Due to Timing Errors



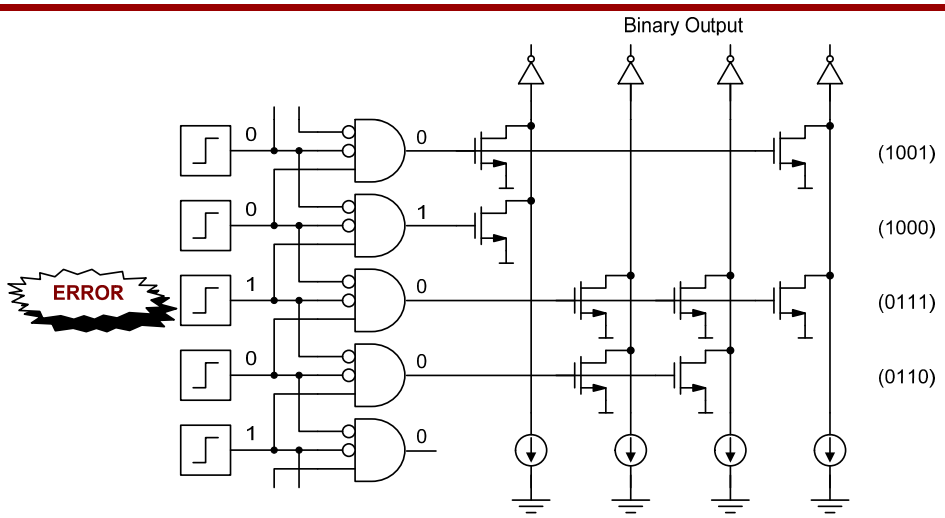
K. Uyttenhove, M.S.J. Steyaert, "A 1.8-V 6-bit 1.3-GHz flash ADC in 0.25-μm CMOS," *Solid-State Circuits, IEEE JSSC*, pp. 1115-1122, July 2003.

Bubble Problem



- Correct output: 1000, actual output: 1110 (!)

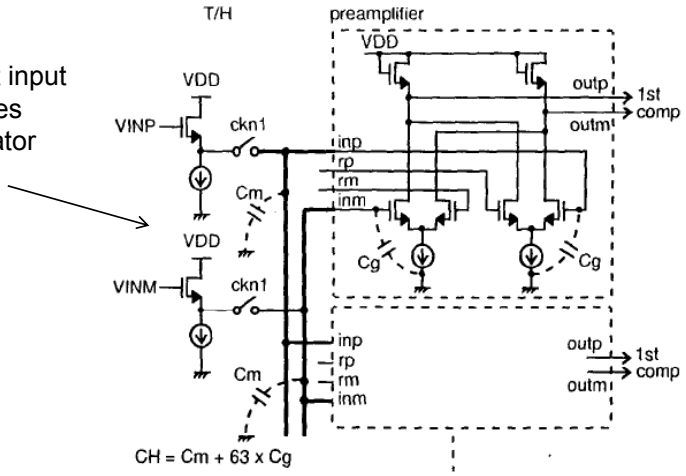
Bubble Tolerant Encoder



- Protects against isolated, single "bubbles"
- Reference: C. W. Mangelsdorf, “A 400-MHz Flash Converter with Error Correction,” IEEE J. Solid-State Ckts., pp. 184-191, Feb. 1990.

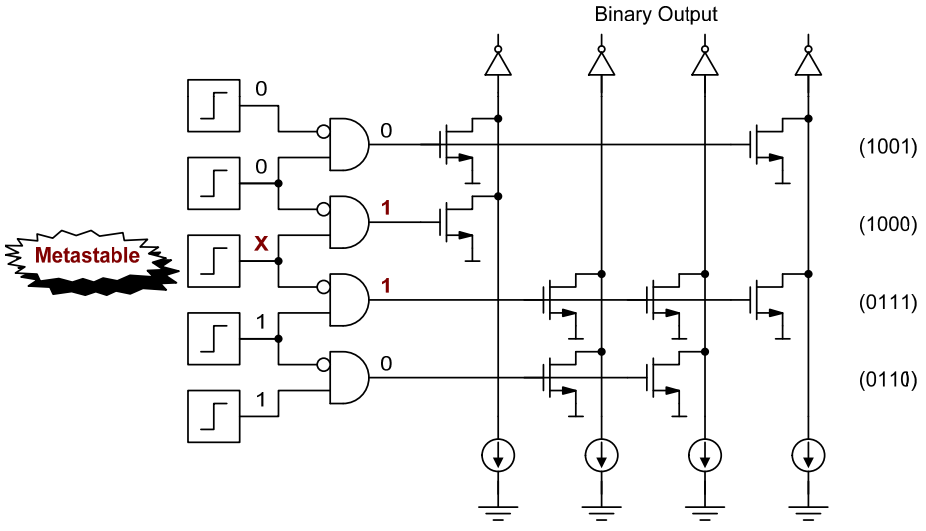
Alternative (or Additional) Solution

Dedicated T/H at input drastically reduces dV/dt at comparator inputs



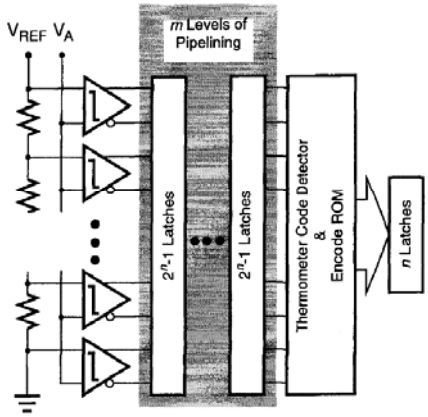
Y. Tamba, and K. Yamakido, "A CMOS 6 b 500 MSample/s ADC for a hard disk drive read channel," ISSCC *Dig. Techn. Papers.* pp.324-325, Feb. 1999.

Metastability



- Different gates interpret metastable output X differently
- Correct output: 0111 or 1000, actual output: 1111

Solution 1: Latch Pipelining

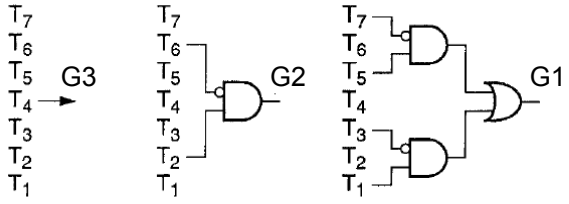


- Use additional latches to create extra gain before generating decoder signals
- Power hungry and area inefficient

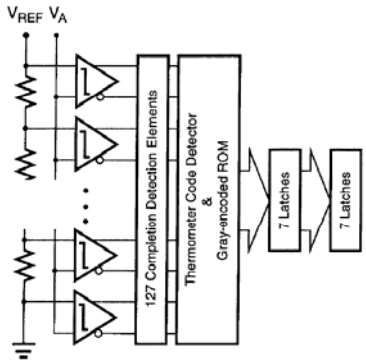
Solution 2: Gray Encoding

Thermometer Code							Gray			Binary		
T_1	T_2	T_3	T_4	T_5	T_6	T_7	G_3	G_2	G_1	B_3	B_2	B_1
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	1
1	**1**	0	0	0	0	0	0	1	1	0	1	0
1	**1**	**1**	0	0	0	0	0	1	0	0	1	1
1	**1**	**1**	**1**	0	0	0	1	1	0	1	0	0
1	**1**	**1**	**1**	**1**	0	0	1	1	1	1	0	1
1	**1**	**1**	**1**	**1**	**1**	0	1	0	1	1	1	0
1	**1**	**1**	**1**	**1**	**1**	**1**	1	0	0	1	1	1

- Each T_i affects only one G_i
  - Avoids disagreement of interpretation by multiple gates
- Also helps protect against sparkles



Efficient Implementation



- Reference
  - C. Portmann and T. Meng, "Power-Efficient Metastability Error Reduction in CMOS Flash A/D Converters," IEEE J. Solid-State Ckts., pp. 1132-40 , Aug. 1996.

Offset

- Typically want offset of each comparator <1/4LSB
  - If we budget half of the input referred offset for the latch, the other half for the pre-amp, this means pre-amp offset must be <1/4LSB / sqrt(2)

$$3\sigma_{VOS} = 3\frac{A_{VT}}{\sqrt{WL}} < \frac{1}{4\sqrt{2}}\frac{FSR}{2^B}$$

- E.g. 6-bit flash ADC, FSR=1V

$$3\frac{A_{VT}}{\sqrt{WL}} < \frac{1}{4\sqrt{2}}\frac{1V}{2^6} = 2.8mV$$

$$WL > \left(\frac{3A_{VT}}{2.8mV}\right)^2 = \left(\frac{3\cdot 4mV\mu m}{2.8mV}\right)^2 = 18.4\mu m^2 \Rightarrow W > \frac{18.4\mu m^2}{0.18\mu m} = 102\mu m$$

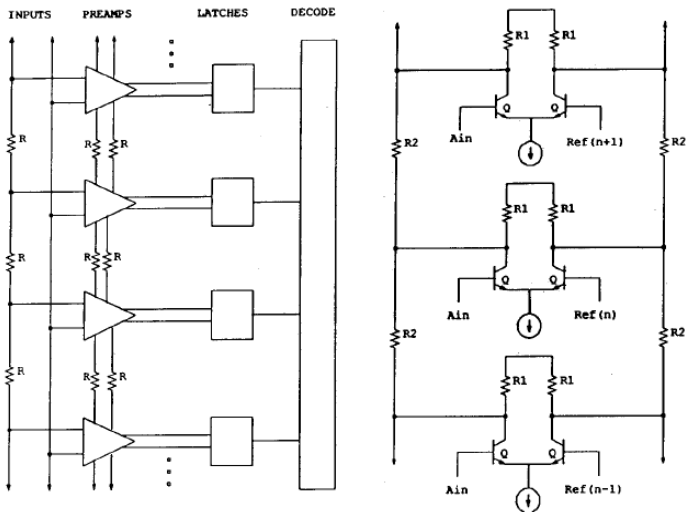
Huge!

Options

- Simply use large devices
  - For each extra bit, need to increase width by 4x, also need to double number of comparators
  - Assuming constant current density, this means each additional bit costs 8x in power!
- Offset cancellation
  - Tends to cost speed
- Offset averaging
- Calibration

Offset Averaging (1)

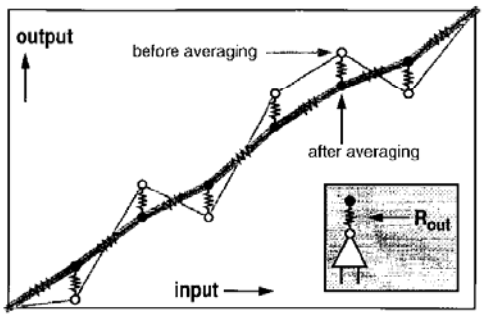
- Output of each differential pair is no longer determined by that pair alone, which results in averaging



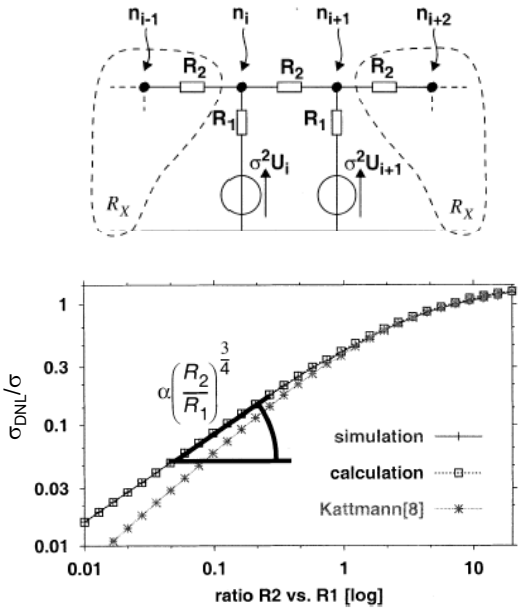
R2/R1=1.3

[Kattmann & Barrow, ISSCC 1991]

Offset Averaging (2)

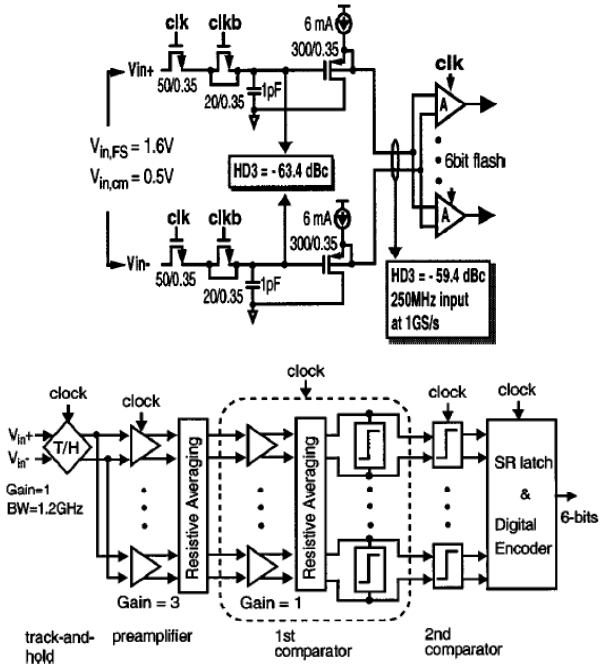


[Bult & Buchwald, JSSC 12/1997]



[Scholtens & Vertregt, JSSC 12/2002]

6-bit Flash ADC with Averaging

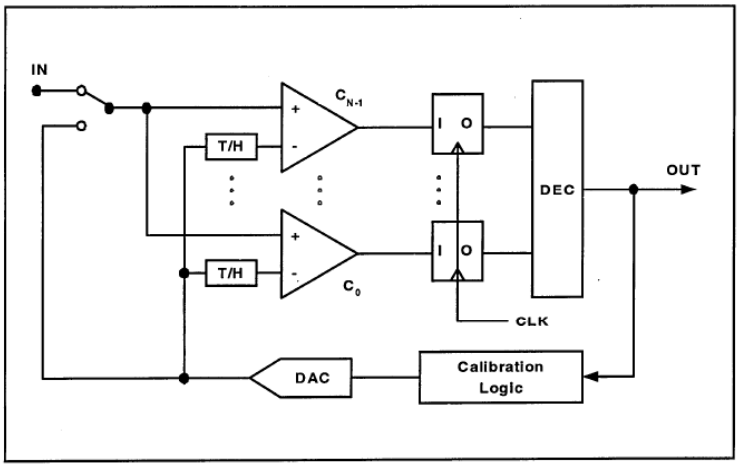


[Choi & Abidi, JSSC 12/2001]

Resolution	6 bits
Conversion Rate	1.3 GS/s
INL/DNL @ Fs = 1 GHz	0.35 LSB / 0.2 LSB
Input Range	1.6 V_{p-p} differential
Input Capacitance (T/H)	1 pF
Bit Error Rate @ fs = 1 GHz	< 10^{-10}
SFDR @ fin = 100 MHz	> 44 dB up to 1.3 GS/s
SNDR @ fin = 630 MHz @ fin = 650 MHz	35 dB @ 1 GS/s 32 dB @ 1.3 GS/s
Power Dissipation (50% due to logic and clock)	500 mW @ 1 GS/s 545 mW @ 1.3 GS/s
Voltage Supply	3.3 V
Active/Total Die Area	2x0.4 mm^2 / 2.2x2.2 mm^2
Technology	0.35-µm CMOS

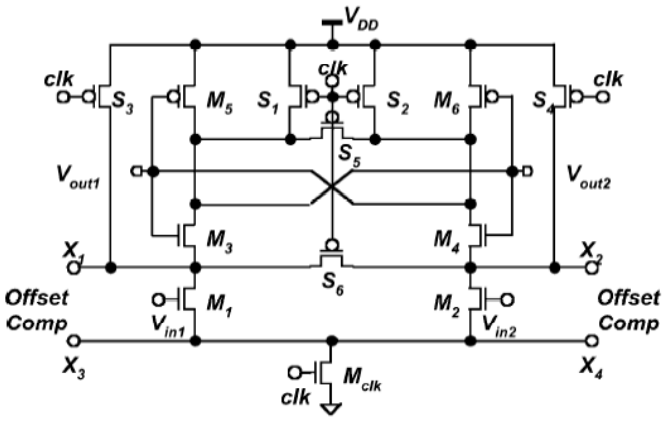
Averaging networks designed to reduce input referred offset by 3x

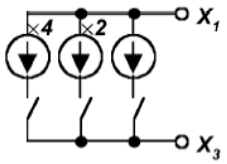
Offset Calibration

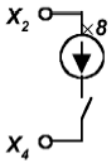


S. Sutardja, "360 Mb/s (400 MHz) 1.1 W 0.35µm CMOS PRML read channels with 6 burst 8-20× over-sampling digital servo," ISSCC Dig. Techn. Papers, Feb. 1999.

Comparator with Integrated Offset DAC

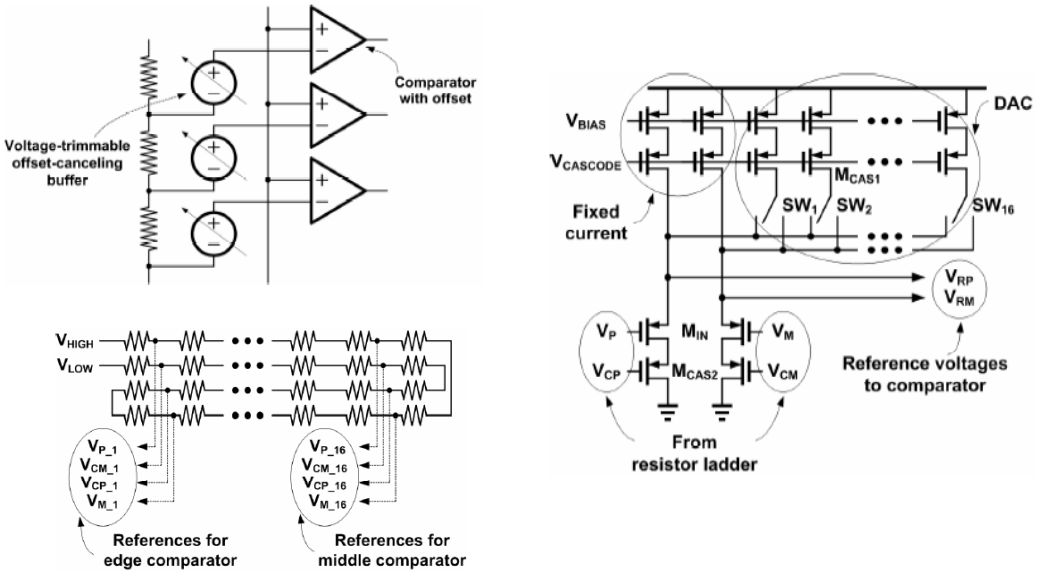






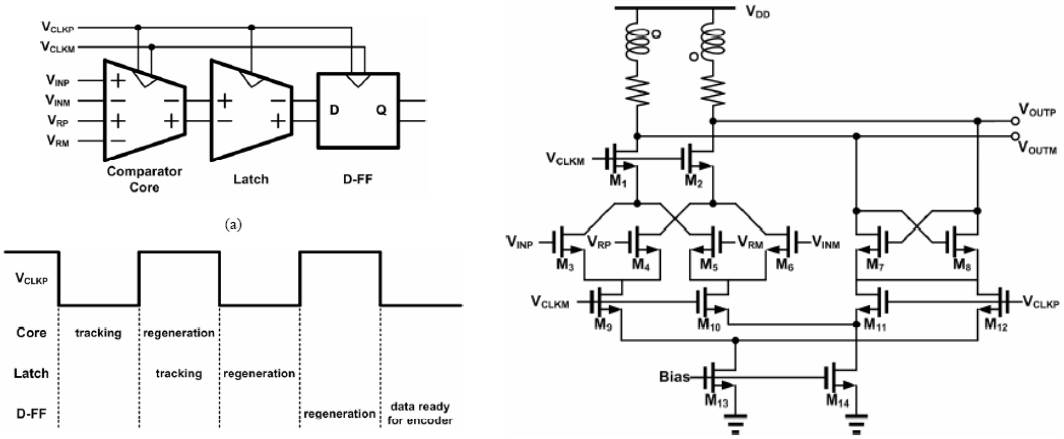
[K.-L.J. Wong and C.-K.K. Yang, "Offset compensation in comparators with minimum input-referred supply noise," JSSC, May 2004.]

High Performance Flash ADC with Calibration (1)



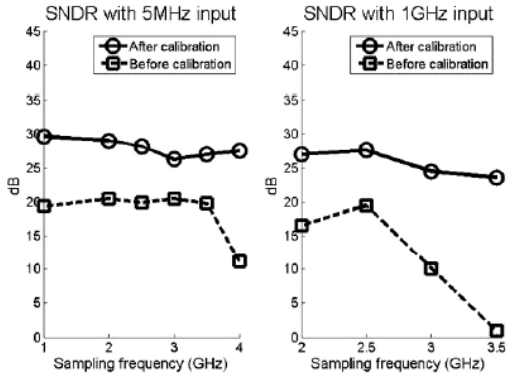
[Park & Flynn, "A 3.5 GS/s 5-b Flash ADC in 90 nm CMOS," CICC 2006]

High Performance Flash ADC with Calibration (2)



[Park & Flynn, "A 3.5 GS/s 5-b Flash ADC in 90 nm CMOS," CICC 2006]

High Performance Flash ADC with Calibration (3)



ADC PERFORMANCE

Technology	CMOS 90 nm
Resolution	5 bits
Supply	1.4 V analog, 1.4 V digital, 1.8 V clock buffer
Input range	± 320 mV (LSB = 20 mV)
Sampling rate	Up to 4 GS/s
Power	227 mW (115 mW: comparators, resistor ladder, bias. 17 mW: D-FFs, encoder, decimator. 95 mW: clock buffer)
DNL @ 3.5 GS/s	-0.83 LSB ~ 0.93 LSB (after calibration) -1.00 LSB ~ 4.51 LSB (before calibration)
INL @ 3.5 GS/s	-0.89 LSB ~ 0.88 LSB (after calibration) -2.20 LSB ~ 1.98 LSB (before calibration)
SNDR	27.5 dB @ 4.0 GS/s, 5 MHz input 23.6 dB @ 3.5 GS/s, 1 GHz input
Active area	0.658 mm^2 (including resistor ladder)
Input capacitance	540 fF
Package	Bare-die on board

[Park & Flynn, "A 3.5 GS/s 5-b Flash ADC in 90 nm CMOS," CICC 2006]

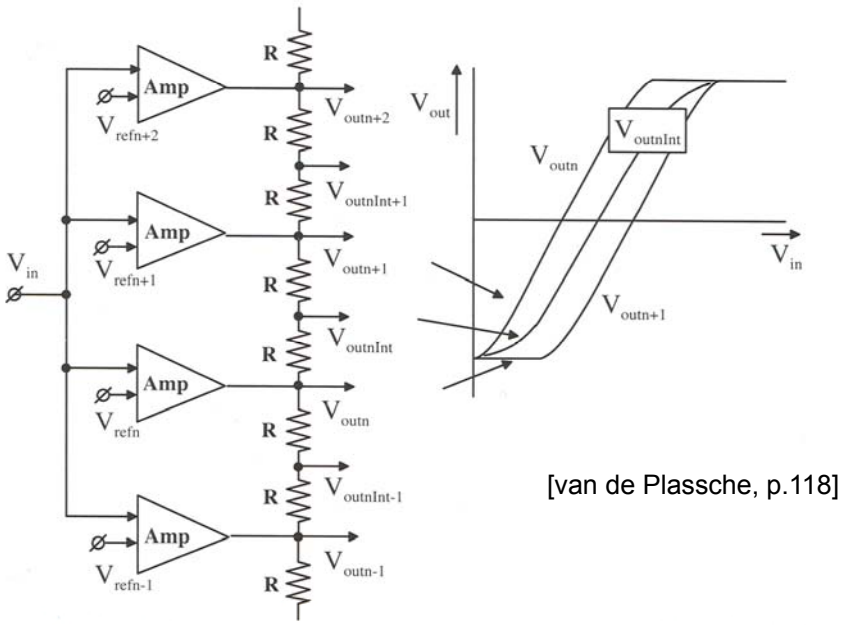
Reducing Complexity

- Example: 10-bit flash ADC
  - Compared to previous 6-bit example we need to
    - Use 16x the number of comparators
    - Increase size & power of each comparator by 16^2 (matching)
  - Input capacitance: $1pF \cdot 16^3 = 4096pF$ (!)
  - Power: $500mW \cdot 16^3 = 2048W$ (!)
- Techniques
  - Interpolation
  - Folding
  - Folding & Interpolation
  - Multi-step conversion, pipelining

Interpolation

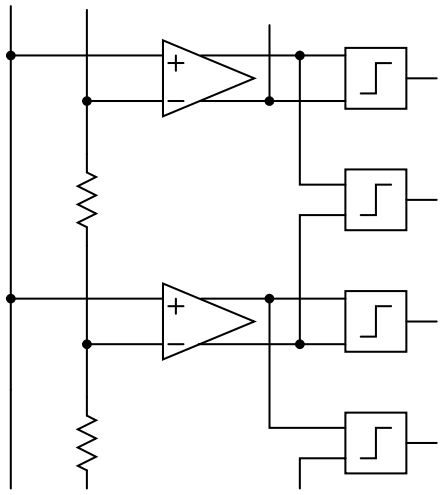
- Idea
  - Interpolation between preamp outputs
- Reduces number of preamps
  - Reduced input capacitance
  - Reduced area, power dissipation
- Same number of latches
- Important “side-benefit”
  - Decreased sensitivity to preamp offset
    - Improved DNL

Concept



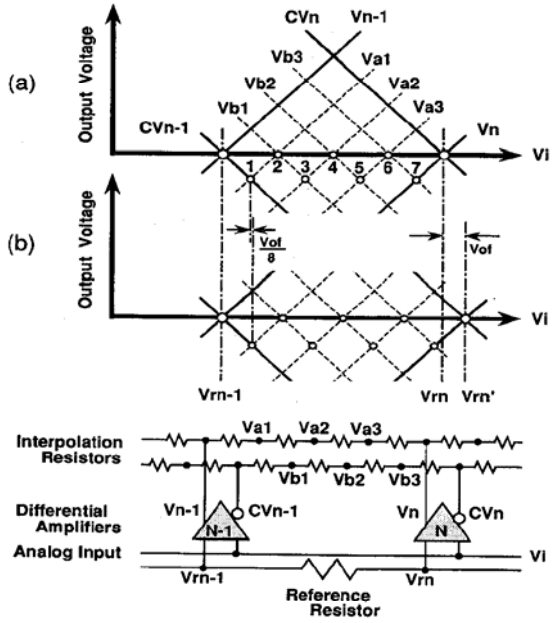
[van de Plassche, p.118]

Differential Implementation



$$\frac{V_A + V_B}{2} = 0 \quad \Leftrightarrow \quad V_A = -V_B$$

Higher Order Interpolation



- Resistors produce additional levels
- Define interpolation factor as ratio ratio of latches and preamps
- The example shown on this slide has M=8

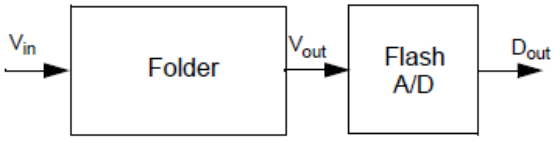
[H. Kimura et al, "A 10-b 300-MHz Interpolated-Parallel A/D Converter," IEEE J. of Solid-State Circuits, pp. 438-446, April 1993.

Potential Issues with Interpolation

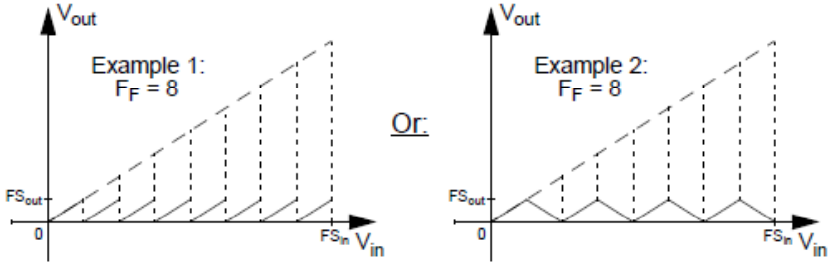
- Must ensure that "linear range" of adjacent preamplifiers overlaps
  - Sets upper bound on preamp gain
- Resistor string reduces signal path bandwidth
  - Sets upper bound on interpolation factor, typically around 4
- For interpolation factors >2, amplifier nonlinearity can limit the precision of zero crossings
  - See e.g. [van de Plassche, p.121]

Folding

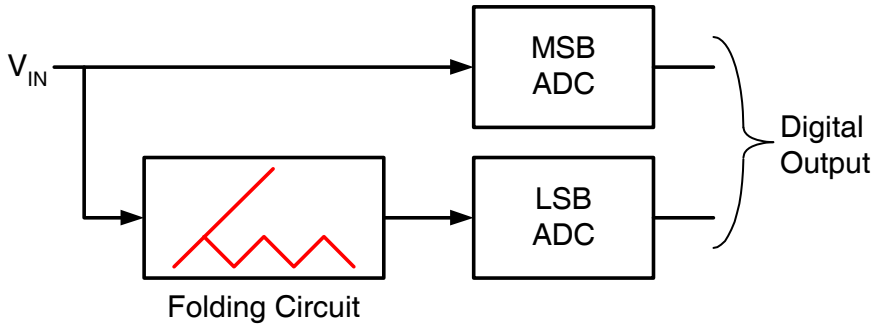
- Insight: most of the comparators in a flash ADC are saturated while only the ones “near” a given input voltage level are required to resolve a small difference.



- The folding operation maps the input signal into successive linear segments, in a periodic fashion (“modulo” operation):

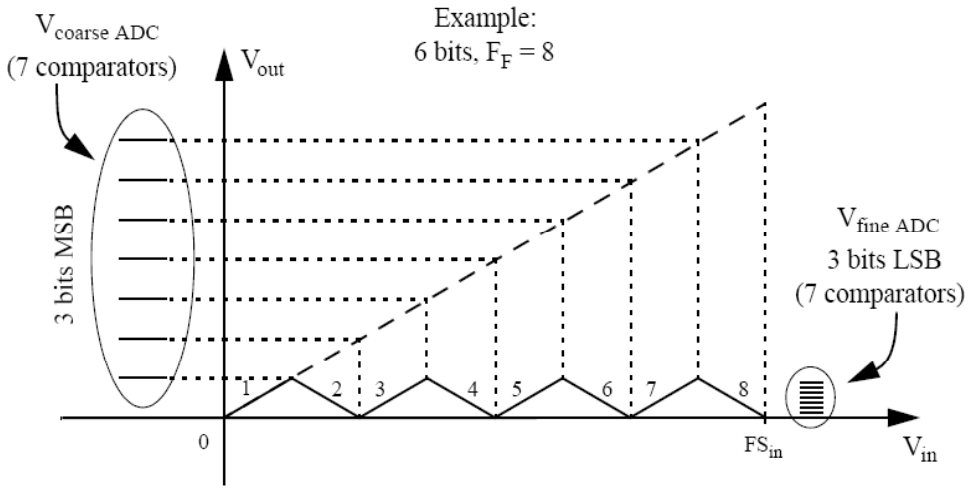


Folding ADC



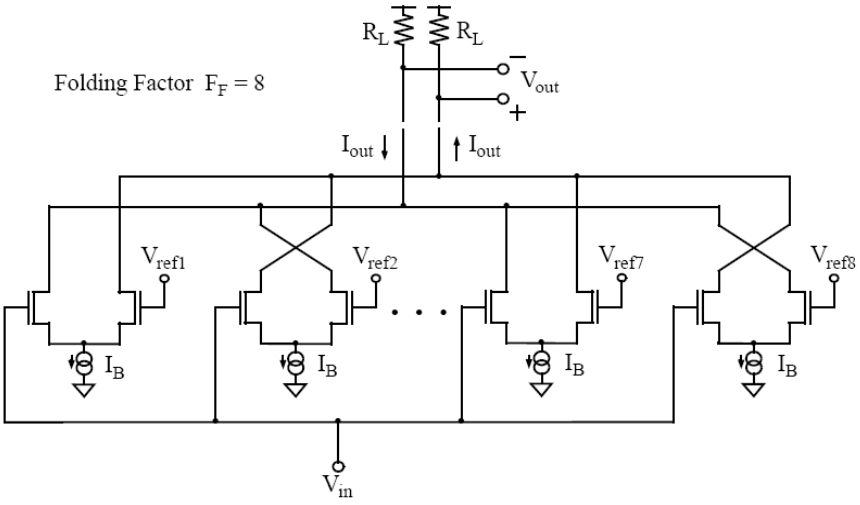
- Fast
- Significantly fewer comparators than flash
- Nonidealities in folder limit attainable resolution to ~10 bits

Example: 6-bit Folding ADC



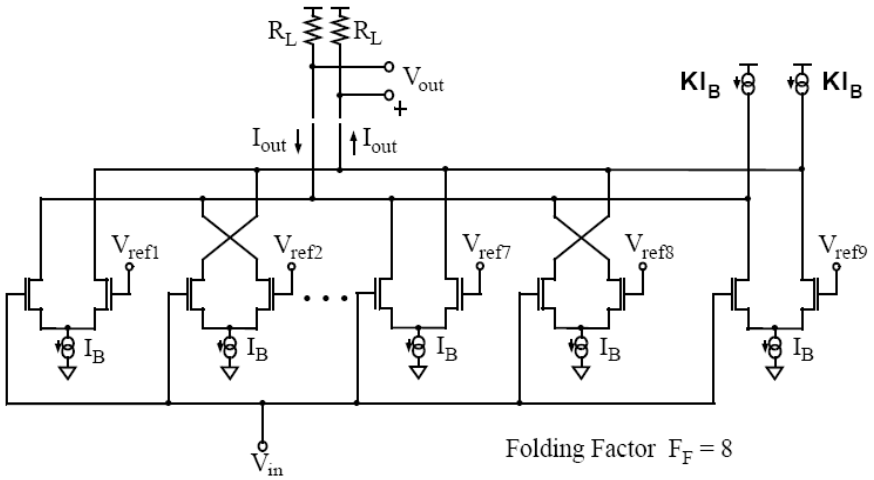
- Coarse ADC determines segment, fine ADC determines level within segment
- Folding factor (F_F) quantifies number of folder output segments

Folder Realization



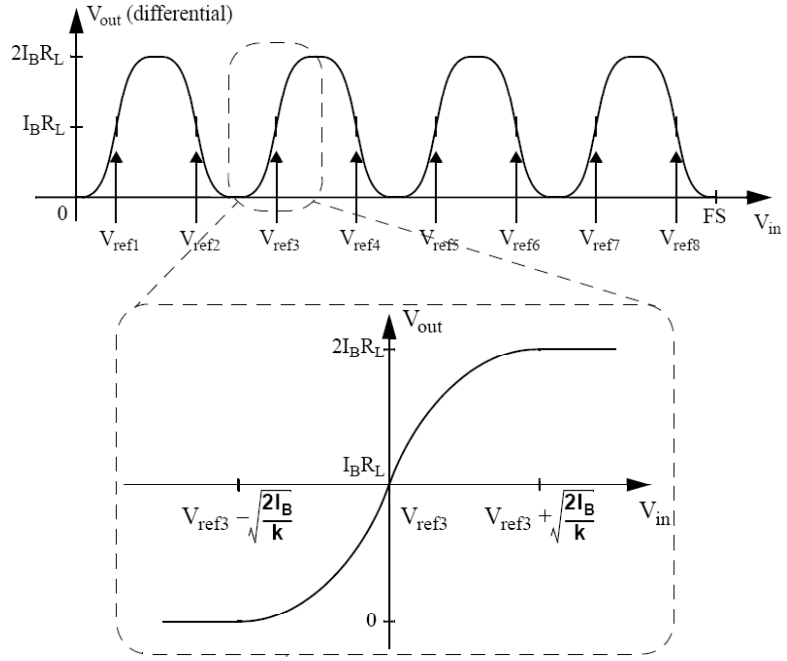
- $V_{ref1} < V_{ref2} < \ldots < V_{ref8}$
- For any V_{in}, only one differential pair is "active", all others are saturated

Improved Realization

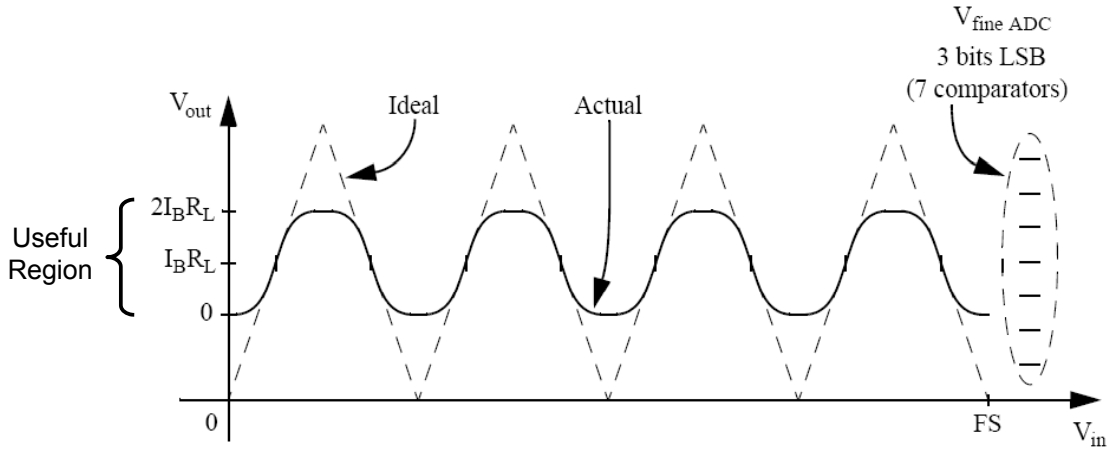


- Extra differential pair removes DC component in V_{out}
- Parameter K controls output common mode

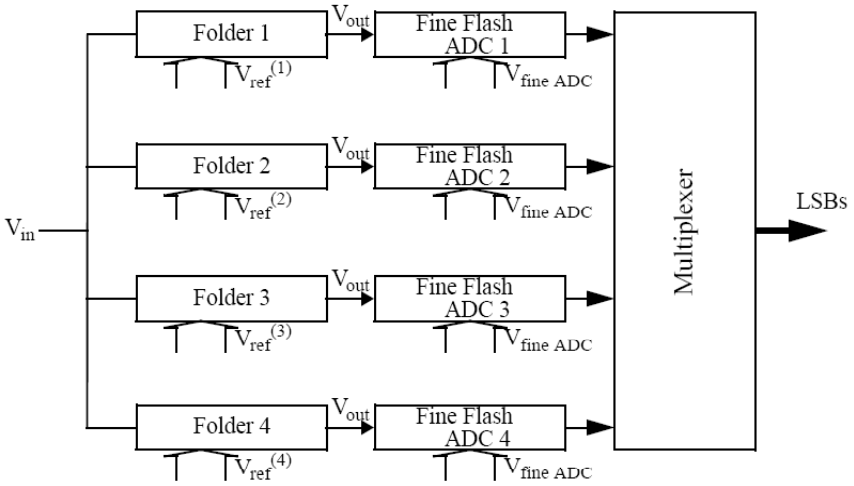
Input-Output Characteristic



Rounding Problem

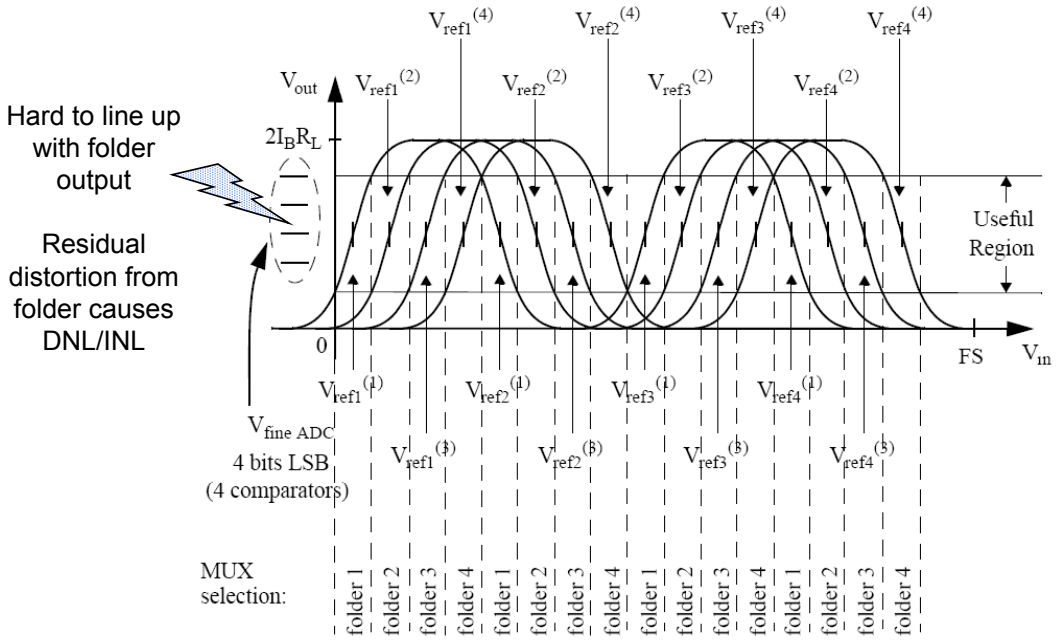


Multiple Folds (1)

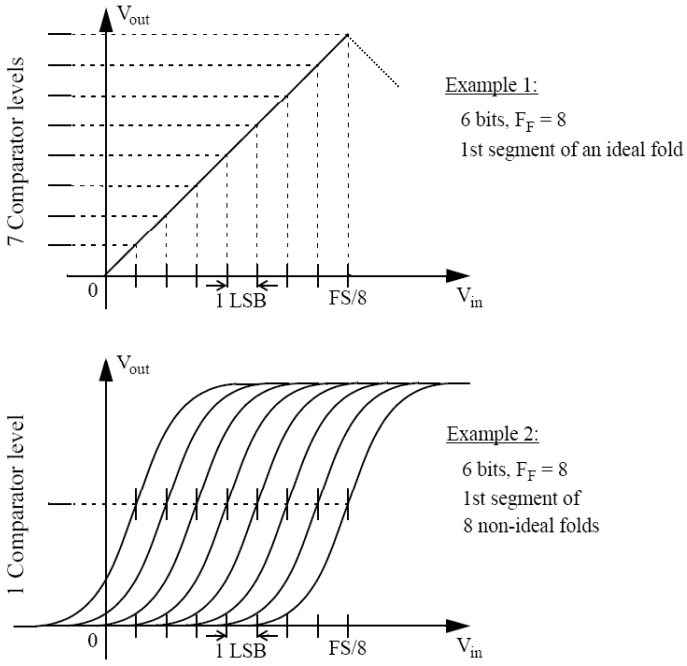


- Idea: Use several folders so that any input value falls into useable "linear" region

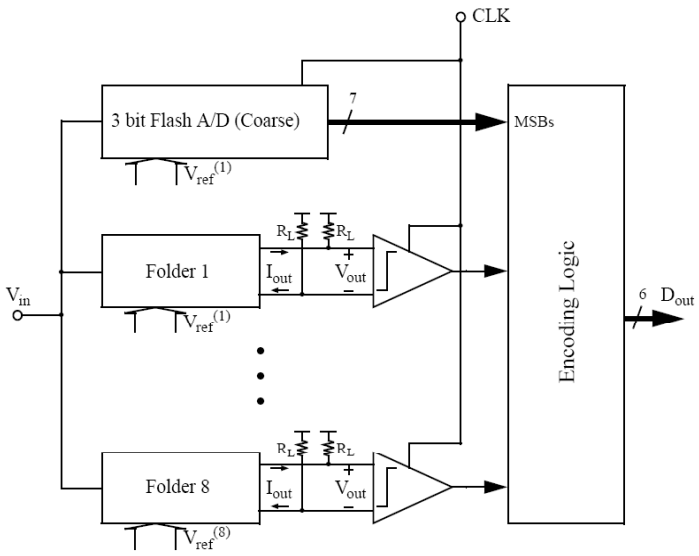
Multiple Folds (2)



Multiple Folds Using Single Threshold (1)

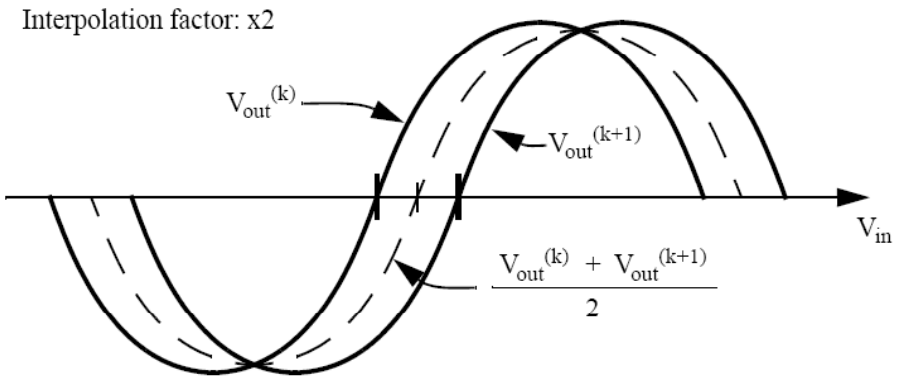


Multiple Folds Using Single Threshold (2)



- Initial idea
  - Use one folder and 7 comparators in LSB section
- Now have
  - 8 Folders and 8 comparators (!)
- Yet another idea
  - Use interpolation to eliminate some of the folders

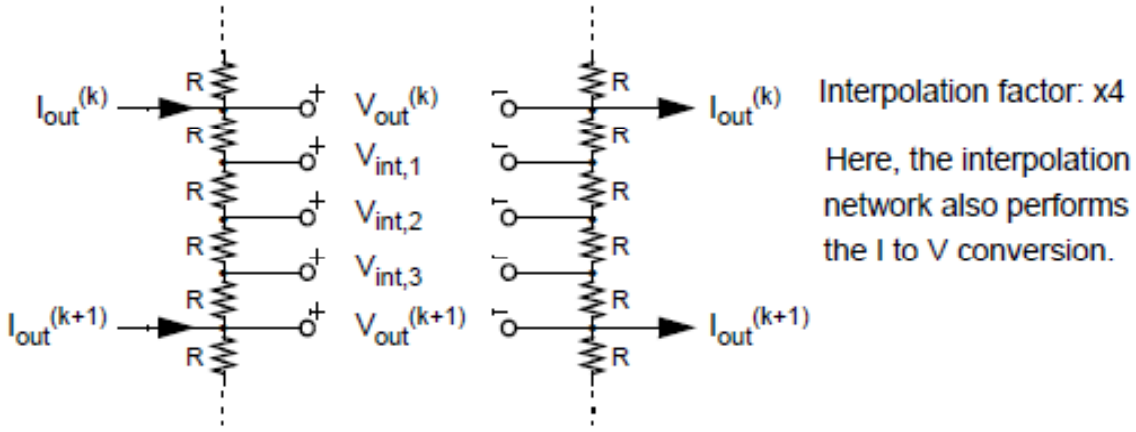
Interpolation



- Same idea as discussed in the context of flash ADCs

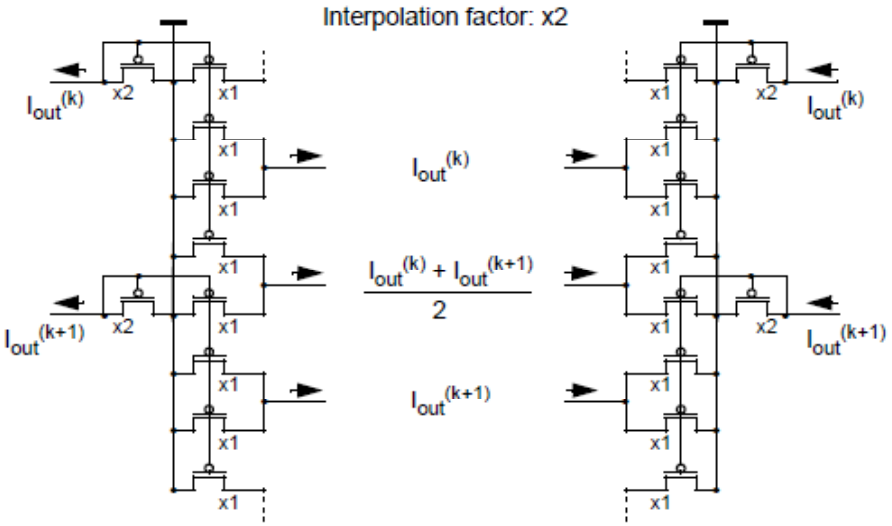
Interpolation Realization (1)

- Resistive interpolation

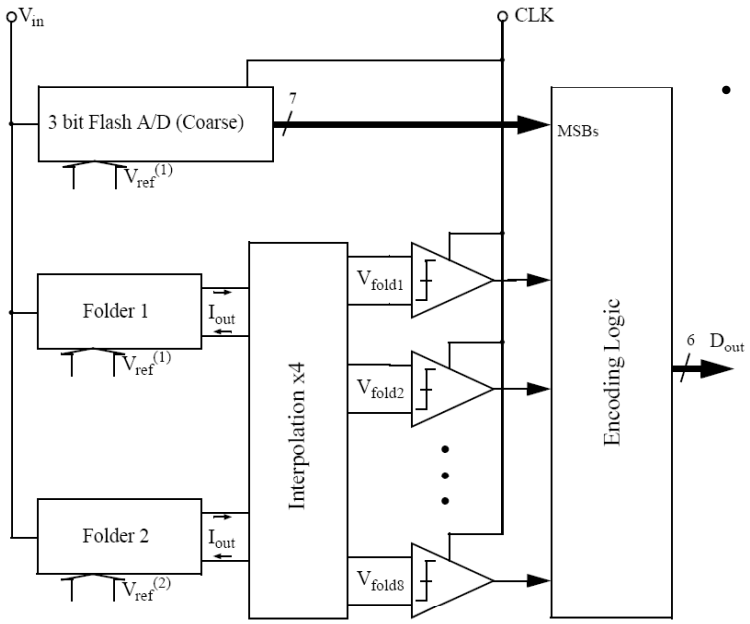


Interpolation Realization (2)

- Current interpolation

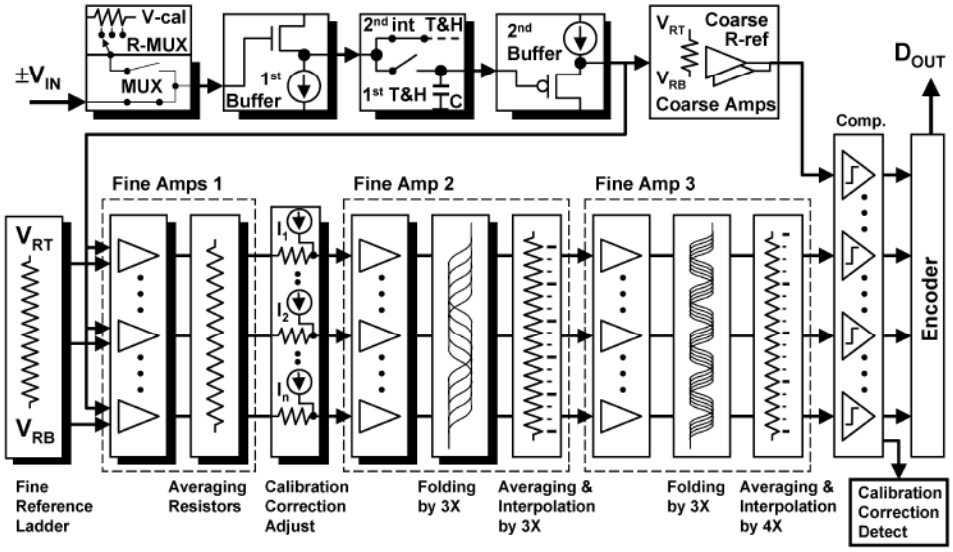


Complete Folding & Interpolating ADC



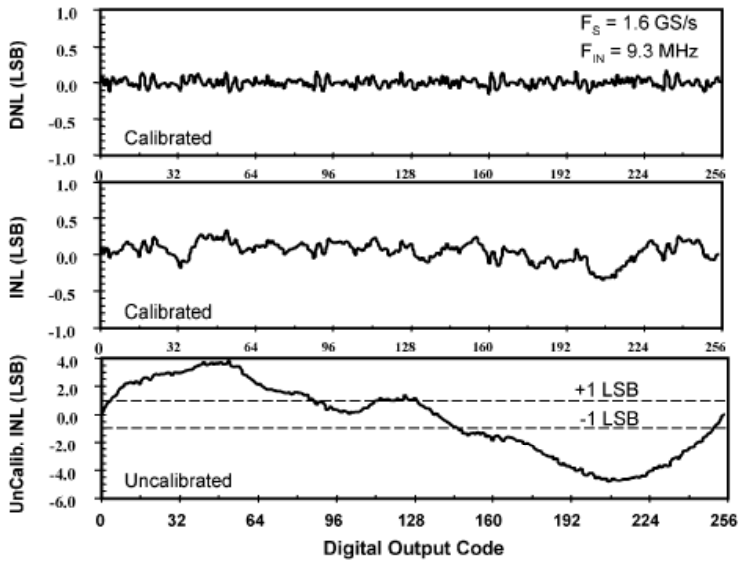
- 6-bit Example
  - 2 folders
  - 4x interpolation
  - F_F=8

State-of-the art Implementation (1)

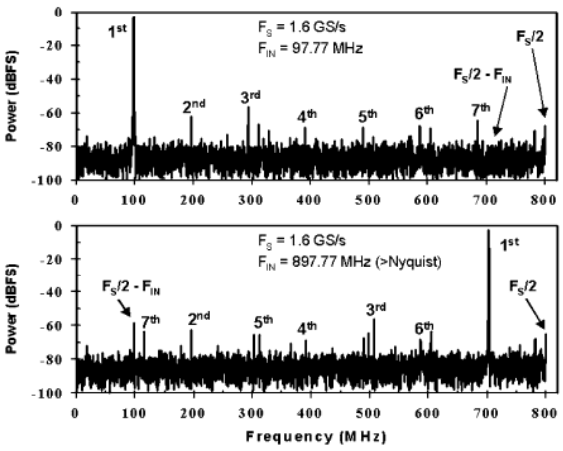


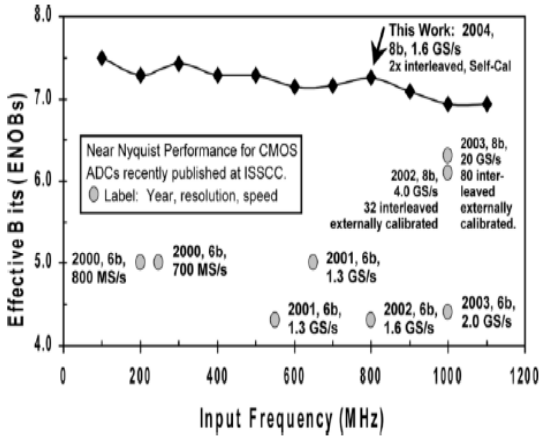
[Taft et al., JSSC 12/2004]

State-of-the art Implementation (2)



State-of-the art Implementation (3)





State-of-the art Implementation (4)

	F_{IN} = 97.77 MHz	F_{IN} = 797.77 MHz (Nyquist)
Sample Rate, F_S	1.6 GS/s	
Resolution	8 bits	
Max DNL	±0.15 LSB	
Max INL	±0.35 LSB	
SNR	48 dB	46 dB
SFDR	61 dB	56 dB
THD	- 57 dB	- 57 dB
ENOBs	7.60	7.26
Interleave aperture offset	< 0.35 ps @ F_S = 1 GS/s & F_{IN} = 1.5 GHz	
Input (–3 dB) Bandwidth	> 1.75 GHz	
Resolution (-0.5 ENOB) Bandwidth	1.0 GHz	
Input Range	±400 mV differential	
Input Capacitance	1.8 pF (to gnd, w/o package)	
Input Termination	50 Ω (100 Ω differential)	
Single Supply	1.8 V	
Analog (DC) Current	245 mA	
Switching (AC) Current	185 mA	
LVDS Output Drivers	90 mA	
ADC Core Power (w/o outputs)	774 mW	
ADC core area	3.6 mm^2	
ADC die area	16 mm^2 (for dual ADC, pad limited)	
Package	128-pin EPQFP	
Technology	0.18 µm CMOS (1-poly, 5-metal) No capacitor module nor dual-gate process	

Folding ADC Problems & Solutions

- Dynamic problems
  - Frequency at the output of a folder is approximately input frequency times folding factor!
    - Finite bandwidth effects can produce zero crossing shifts
  - Delay through coarse/fine signal path is not well matched
- Possible solution
  - Add track & hold circuit at ADC input
    - This was done in the implementation shown in slides 20-23
- Static problems
  - Offsets in folder transistors can cause DNL, INL
  - Interpolation with a factor greater 2x can introduce DNL, INL due to amplifier nonlinearity
- Possible solutions
  - Averaging, calibration

Selected References (1)

Flash ADCs, Offset Averaging

1. K. Kattmann and J. Barrow, "A Technique for Reducing Differential Non-Linearity Errors in Flash A/D Converters," *ISSCC Digest of Technical Papers*, pp. 170-171, Feb. 1991.
2. K. Bult and A. Buchwald, "An embedded 240-mW 10-b 50-MS/s CMOS ADC in 1-mm^2," *IEEE J. of Solid-State Ckts.,* pp. 1887-1895, Dec. 1997.
3. M. Choi and A. A. Abidi, "A 6 b 1.3 Gsample/s A/D converter in 0.35μm CMOS," *IEEE J. Solid-State Circuits*, pp. 1847–1858, Dec. 2001.
4. C. S. Scholtens and M. Vertregt, "A 6-b 1.6-Gsample/s Flash ADC in 0.18-μm CMOS Using Averaging Termination," *IEEE J. of Solid-State Ckts.*, vol. 37, pp. 1599-1609, Dec. 2002.
5. X. Jiang, M-C. F. Chang, "A 1-GHz signal bandwidth 6-bit CMOS ADC with power-efficient averaging," *IEEE J. of Solid-State Circuits*, pp. 532- 535, Feb. 2005.

Folding and Interpolating A/D Converters

6. R.C. Taft et al., "A 1.8-V 1.6-GSample/s 8-b self-calibrating folding ADC with 7.26 ENOB at Nyquist frequency," IEEE J. Solid-State Ckts., pp. 2107-2115 Dec. 2004.
7. B. Nauta and A. G. W. Venes, "A 70-MS/s 110-mW 8-b CMOS Folding and Interpolating A/D Converter," *IEEE J. of Solid-State Circuits*, pp. 1302-1308, Dec. 1995.

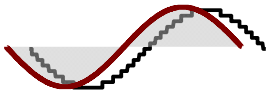
Selected References (2)

8. M. P. Flynn and B. Sheahan, “A 400-MSample/s, 6-b CMOS Folding and Interpolating ADC,” *IEEE J. of Solid-State Circuits*, pp. 1932-1938, Dec. 1998.
9. H. Pan, M. Segami, M. Choi, J. Cao, F. Hatori and A. Abidi, “A 3.3V, 12b, 50MSample/s A/D converter in 0.6mm CMOS with over 80dB SFDR,” *ISSCC Digest of Technical Papers*, pp. 40-41, Feb. 2000.
10. M.-J. Choe, B.-S. Song and K. Bacrania, “A 13b 40MSample/s CMOS pipelined folding ADC with background offset trimming,” *ISSCC Digest of Technical Papers*, pp. 36-37, Feb. 2000.
11. G. Hoogzaad and R. Roovers, “A 65-mW, 10-bit, 40-Msample/s BiCMOS Nyquist ADC in 0.8 mm2,” *IEEE J. Solid-State Circuits*, pp. 1796-1802, Dec. 1999.
12. K. Nagaraj, F. Chen, T. Le and T. R. Viswanathan, “Efficient 6-bit A/D converter using a 1-bit folding front end,” *IEEE J. Solid-State Circuits*, pp. 1056-1062, Aug. 1999.
13. M.-J. Choe and B.-S. Song, “An 8b 100MSample/s CMOS pipelined folding ADC,” *VLSI Symposium Digest of Technical Papers*, pp. 81-82, Jun. 1999.
14. K. Bult and A. Buchwald, “An embedded 240-mW 10-b 50-MS/s CMOS ADC in 1 mm2,” IEEE J. Solid-State Circuits, vol. 32, pp. 1887-1895, Dec. 1997.
15. P. Vorenkamp and R. Roovers, “A 12-b, 60-MSample/s cascaded folding and interpolating ADC,” IEEE J. Solid-State Circuits, vol. 32, pp. 1876-1886, Dec. 1997.
16. A. G. W. Venes and R. J. van de Plassche, “An 80-MHz, 8-b CMOS folding A/D converter with distributed track-and-hold preprocessing,” IEEE J. Solid-State Circuits, vol. 31, pp. 1846-1853, Dec. 1996.

Selected References (3)

17. M. P. Flynn and D. J. Allstot, “CMOS folding A/D converters with current-mode interpolation,” IEEE J. Solid-State Circuits, vol. 31, pp. 1248-1257, Dec. 1996.
18. R. Roovers and M. S. J. Steyaert, “A 175 Ms/s, 6 b, 160 mW, 3.3 V CMOS A/D converter,” IEEE J. Solid-State Circuits, vol. 31, pp. 938-944, Jul. 1996.
19. W. T. Colleran and A. A. Abidi, “A 10-b, 75-MHz two-stage pipelined bipolar A/D converter,” IEEE J. Solid-State Circuits, vol. 28, pp. 1187-1199, Dec. 1993.
20. J. van Valburg and R. J. van de Plassche, “An 8-b 650-MHz folding ADC,” IEEE J. Solid-State Circuits, vol. 27, pp. 1662-1666, Dec. 1992.
21. R. J. van de Plassche and P. Baltus, “An 8-bit 100-MHz full-Nyquist analog-to-digital converter,” IEEE J. Solid-State Circuits, vol. 23, pp. 1334-1344, Dec. 1988.
22. R. E. J. Van de Grift, I. W. J. M. Rutten and M. van der Veen, “An 8-bit video ADC incorporating folding and interpolation techniques,” IEEE J. Solid-State Circuits, vol. SC-22, pp. 944-953, Dec. 1987.
23. R. E. J. van de Grift and R. J. van de Plassche, “A monolithic 8-bit video A/D converter,” IEEE J. Solid-State Circuits, vol. SC-19, pp. 374-378, Jun. 1984.
24. R.C. Taft, et al., "A 1.8V 1.0GS/s 10b self-calibrating unified-folding-interpolating ADC with 9.1 ENOB at Nyquist frequency," *ISSCC Digest of Technical Papers*, pp.78-79, 79a, Feb. 2009.

Pipeline ADCs



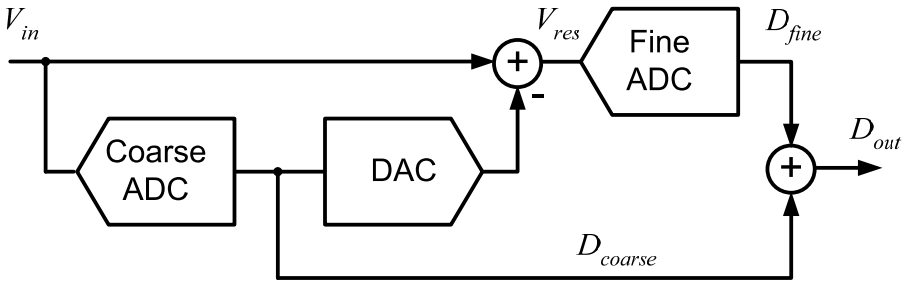
Katelijn Vleugels
Stanford University

Outline

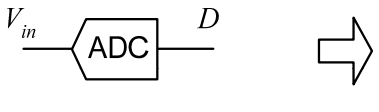
- General idea of multi-step A/D conversion
- Pipeline ADC basics
  - Ideal block diagram and operation, impact of block nonidealities
- Ways to deal with nonidealities
  - Redundancy, calibration
- CMOS implementation details
  - Stage scaling, MDAC design

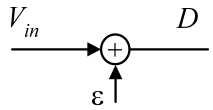
General Concept of Multi-Step Conversion

- General idea (two-step example)
  1. Perform a "coarse" quantization of the input
  2. Compute residuum (error) of step 1 conversion using a DAC and subtractor
  3. Digitize computed residuum using a second "fine" quantizer and digitally add to output

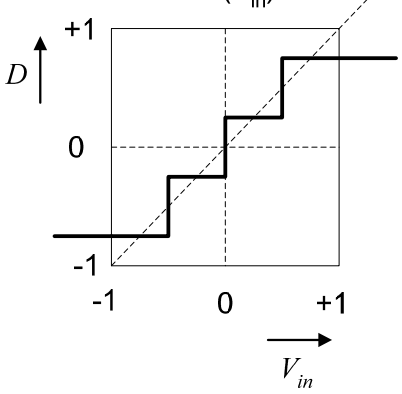


Quantizer Model

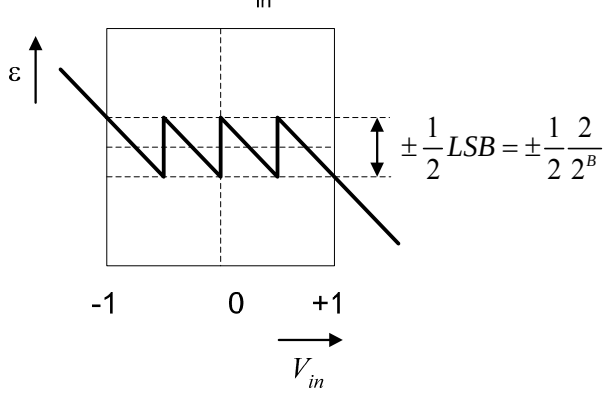




Transfer function
$D = Q(V_{in})$

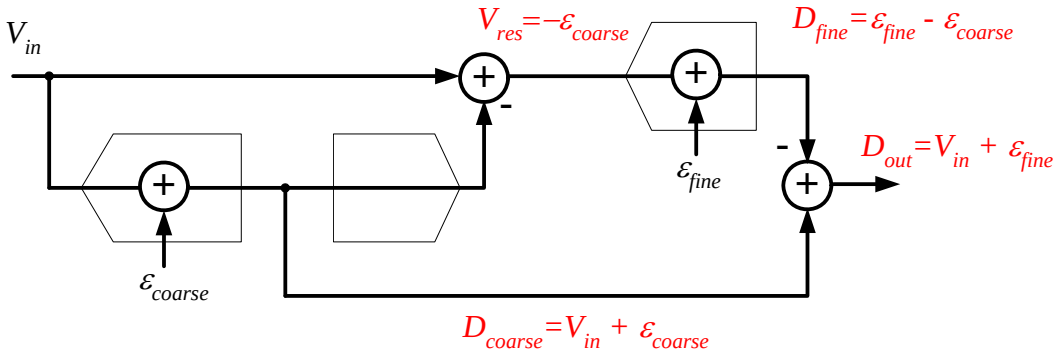


Quantization Error
$\varepsilon = D - V_{in}$



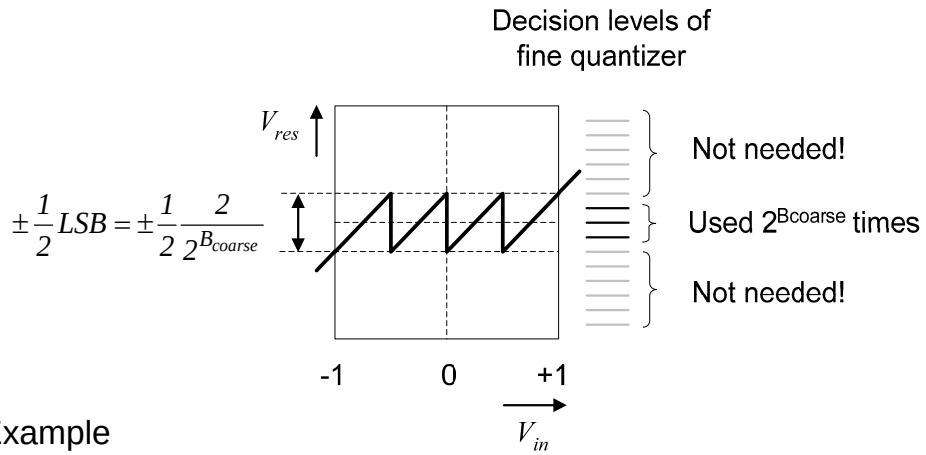
Analysis

- Assuming ideal DAC (for now)



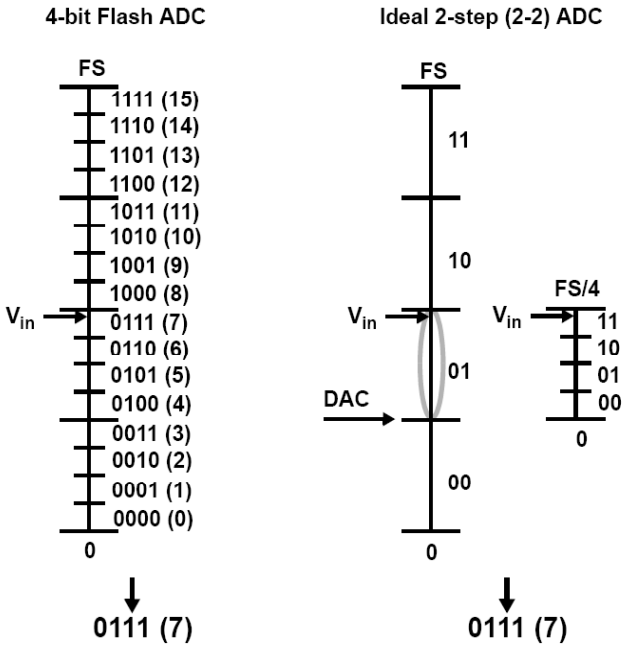
- Output contains only quantization error from fine ADC!

Input to Fine Quantizer (V_{res})

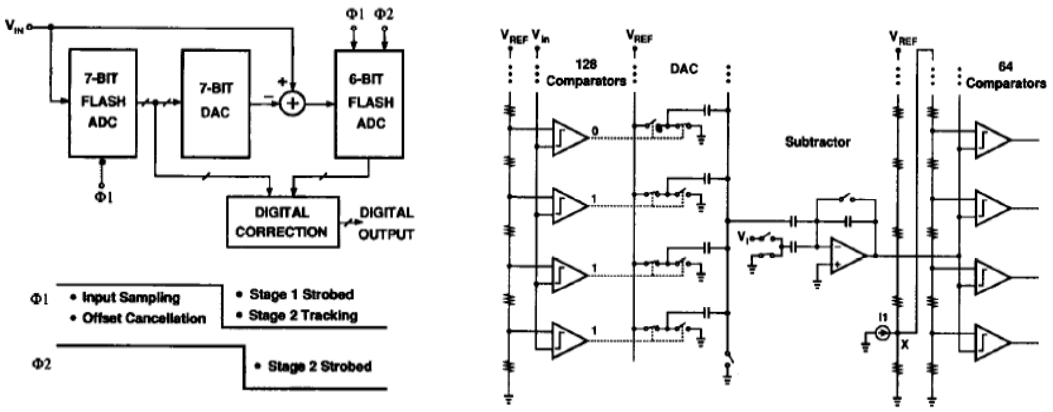


- Example
  - Three decision levels in coarse and fine quantizer
  - Aggregate ADC resolution is 4 bits (3+4·3 decision levels)
  - Need only 6 comparators, compared to 15 in a 4-bit flash ADC
    - Advantage becomes more pronounced at higher resolutions

Alternative Illustration



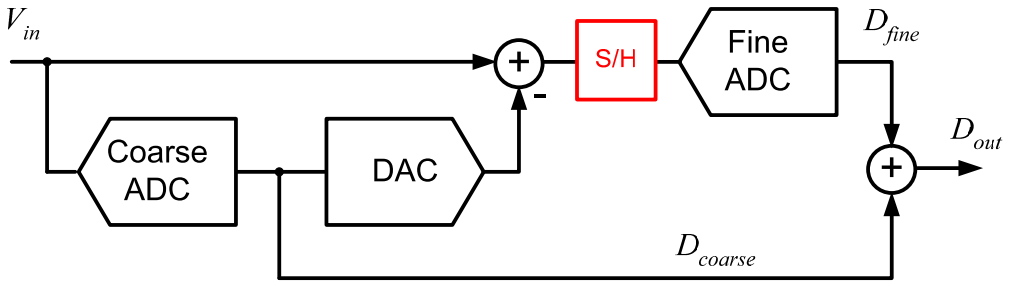
Two-Step ADC Example



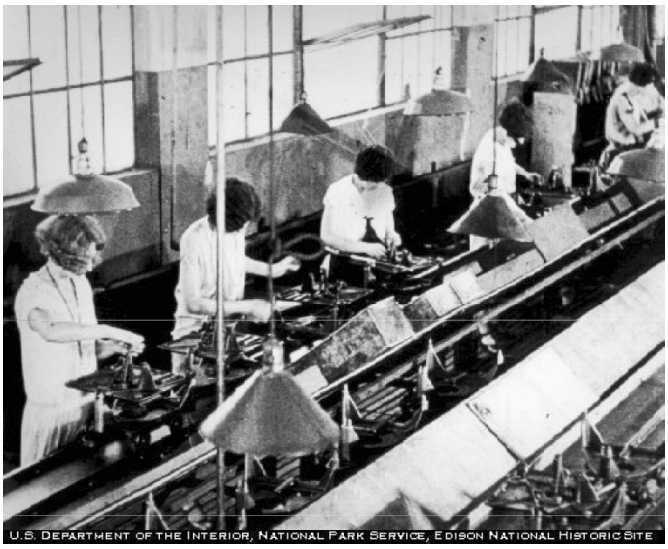
B. Razavi and B.A. Wooley, "A 12-b 5-Msample/s two-step CMOS A/D converter," IEEE JSSC, pp. 1667-1678, Dec. 1992.

Limitations (1)

- Conversion time is proportional to number of stages employed
  - E.g. for a two-step ADC, time required for conversion is $T_{conv} = 2 \cdot T_{A/D} + T_{D/A} + T_{SUB}$
- Solution
  - Introduce a sample and hold operation after subtraction
  - Fine ADC has one full clock cycle until new residuum becomes available
  - "Pipelining"

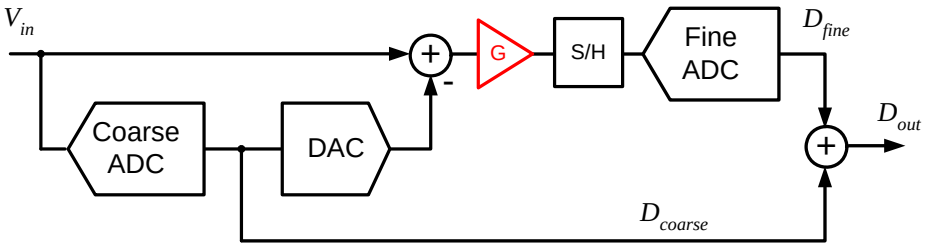


Pipelining – An Old Idea

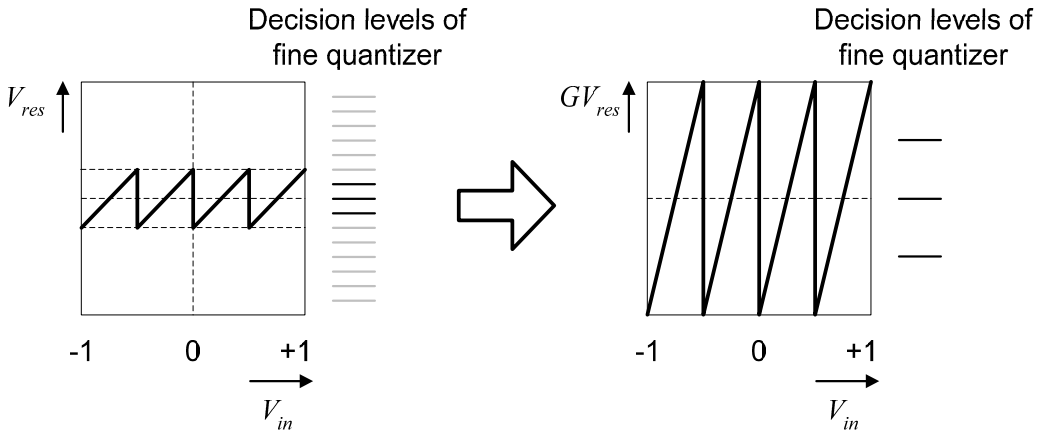


Limitations (2)

- Fine ADC(s) must have precision commensurate with overall target resolution
  - E.g. 8-bit converter with 4-bit/4-bit partition; fine 4-bit decision levels must have "8-bit precision"
- Solution
  - Introduce gain after subtraction

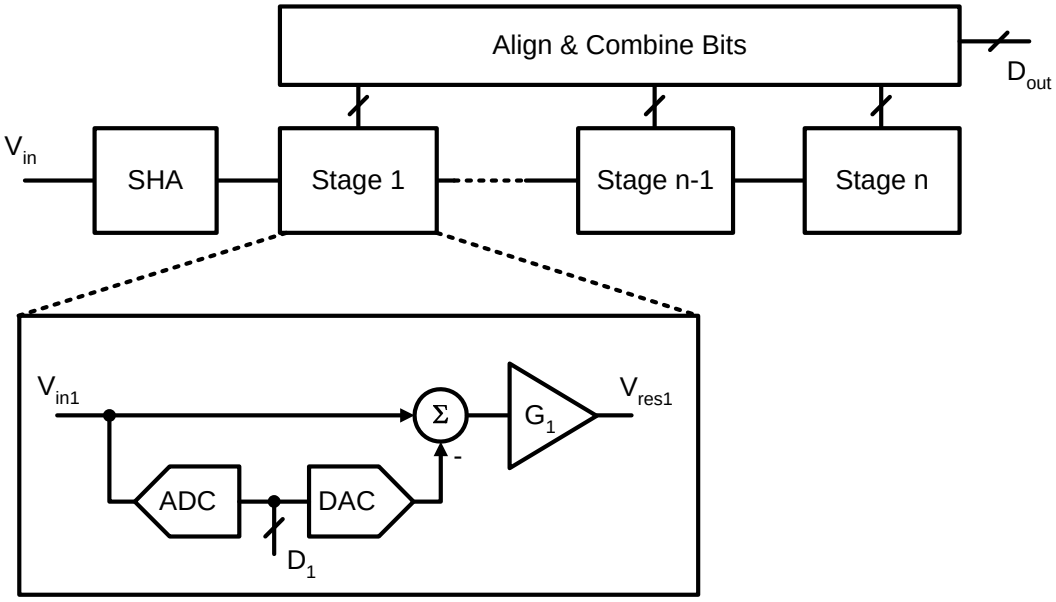


Input to Fine Quantizer with Gain

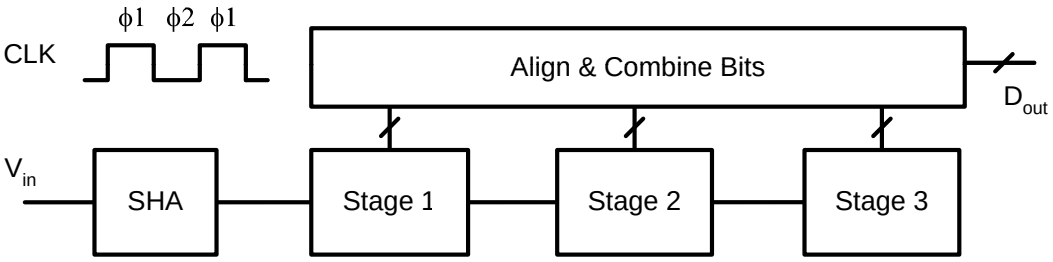


- No longer need precision comparators

Pipeline ADC Block Diagram



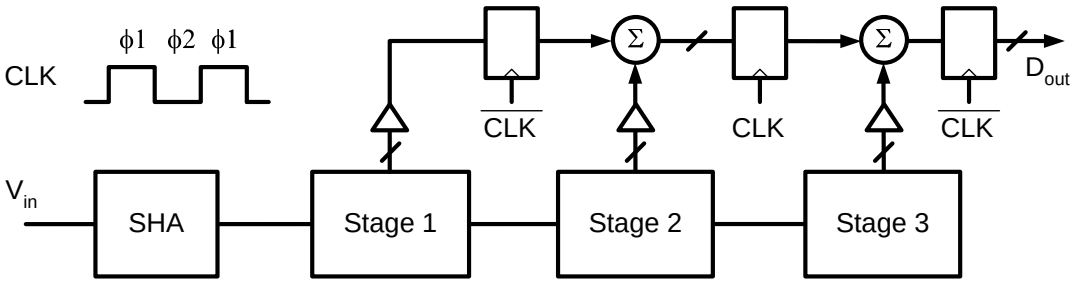
Concurrent Stage Operation



ϕ1	ACQUIRE	CONVERT	ACQUIRE	CONVERT
ϕ2	BUFFER	ACQUIRE	CONVERT	ACQUIRE

- Stages operate on the input signal like a shift register
- New output data every clock cycle, but each stage introduces ½ clock cycle latency

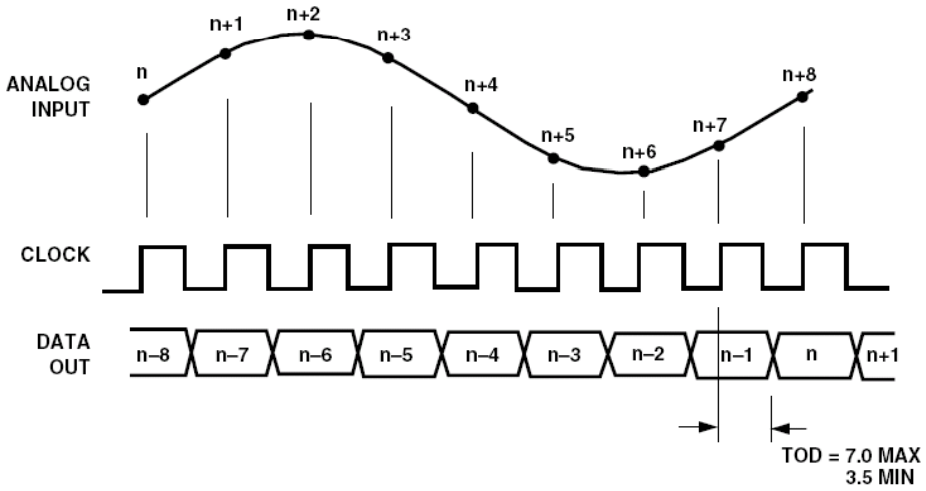
Data Alignment



ϕ1	ACQUIRE	CONVERT	ACQUIRE	CONVERT
ϕ2	BUFFER	ACQUIRE	CONVERT	ACQUIRE

- Digital shift register aligns sub-conversion results in time
- Digital output is taken as weighted sum of stage bits

Latency



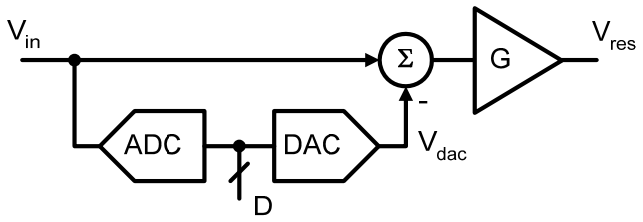
[Analog Devices, AD9226 Data Sheet]

Pipeline ADC Characteristics

- Number of components grows linearly with resolution
  - Unlike flash ADC, where components ~ 2^B
- Pipeline ADC trades latency for conversion speed
  - Throughput limited by speed of one stage
    - Enables high-speed operation
  - Latency can be an issue in some applications
    - E.g. in feedback control loops
- Pipelining only possible with good analog "memory elements"
  - Calls for implementation in CMOS using switched-capacitor circuits

Stage Analysis

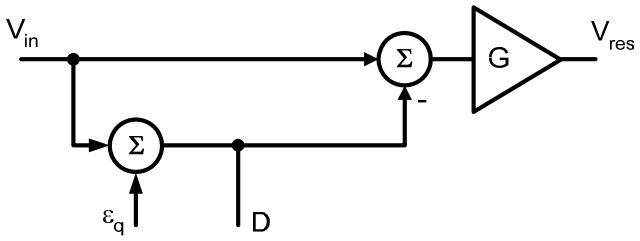
- Ignore timing/clock delays for simplicity



$$D = Q(V_{in})$$

$$V_{res} = G \cdot [V_{in} - V_{dac}]$$

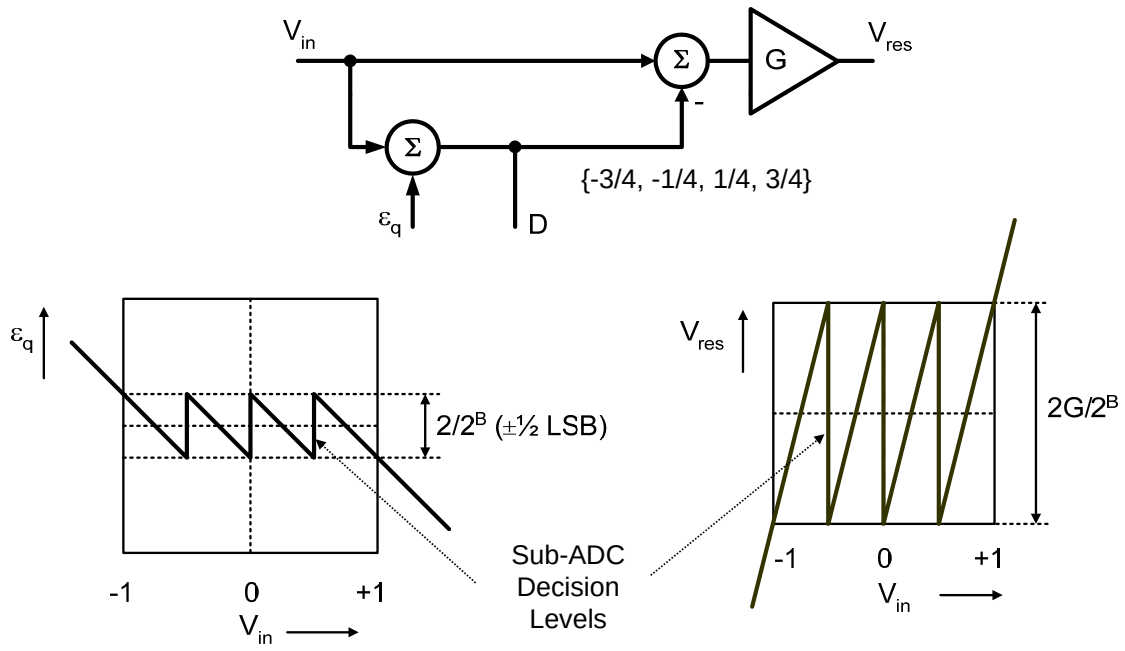
Stage Model with Ideal DAC



$$D = V_{in} + \varepsilon_q \qquad V_{res} = -G \cdot \varepsilon_q$$

- Residue of pipeline stage (V_{res}) is equal to (-gain) times sub-ADC quantization error

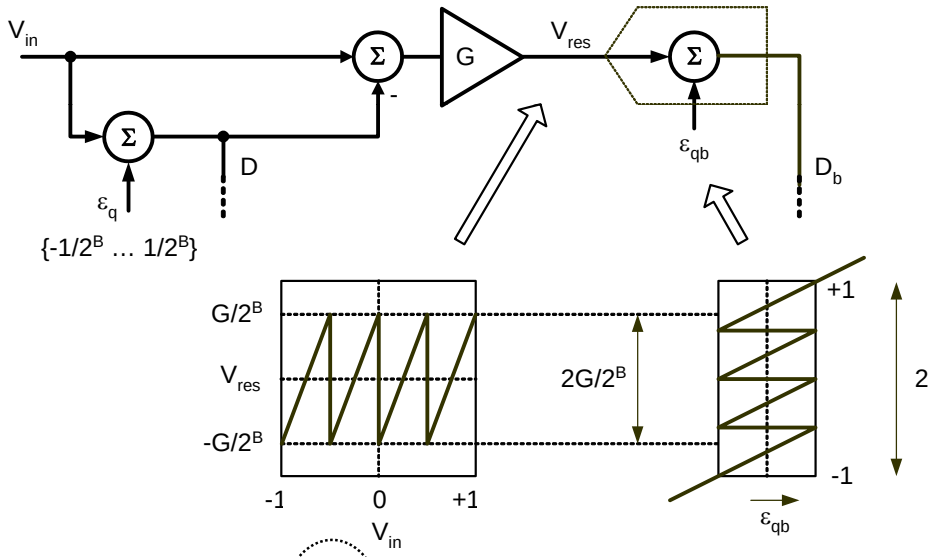
"Residue Plot" (2-bit Sub-ADC)



Questions

- How to pick stage gain G for a given sub-ADC resolution?
- Impact and compensation of nonidealities?
  - Sub-ADC errors
  - Amplifier offset
  - Amplifier gain error
  - Sub-DAC error
- Begin to explore these questions using a simple example
  - First stage with 2-bit sub-ADC, followed by 2-bit backend

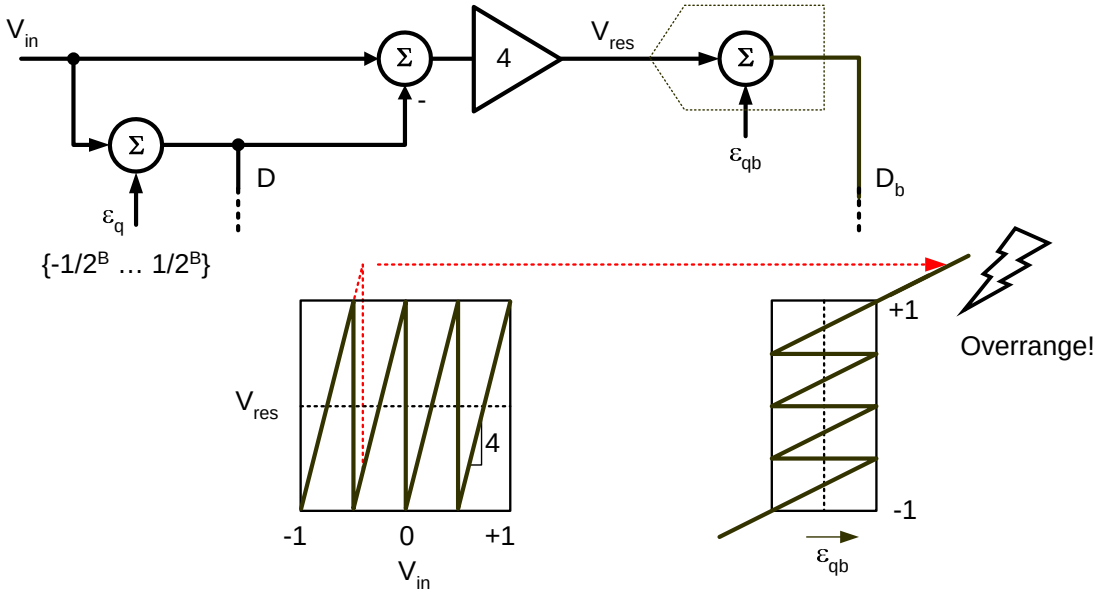
Upper Bound for Stage Gain



$$D_{out} = V_{in} + \frac{\varepsilon_{qb}}{G}$$

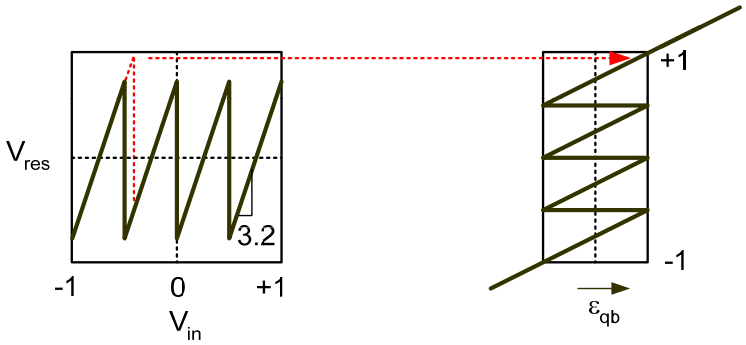
Grows out of ±½ LSB bounds for $G>2^B$

Issue with G=2^B



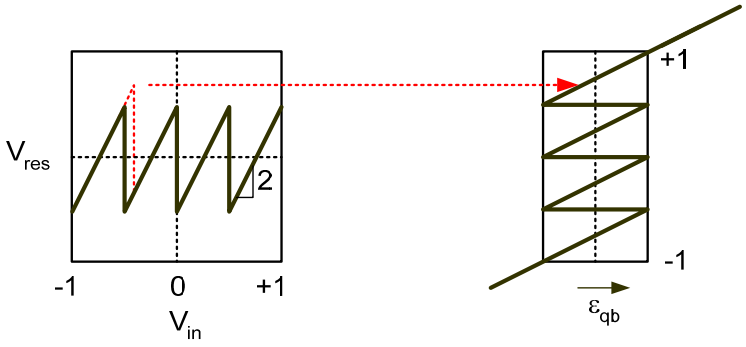
- Any error in sub-ADC decision levels will overload backend ADC and thereby deteriorate ADC transfer function

Idea #1: G slightly less than 2^B



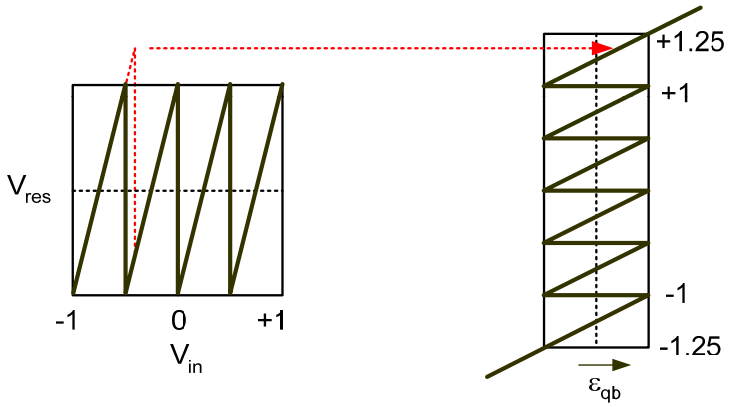
- Effective stage resolution can be non-integer ($R=\log_2 G$)
  - E.g. $R = \log_2 3.2 = 1.68$ bits
- See e.g. [Karanicolas 1993]

Idea #2: G < 2^B, but Power of Two



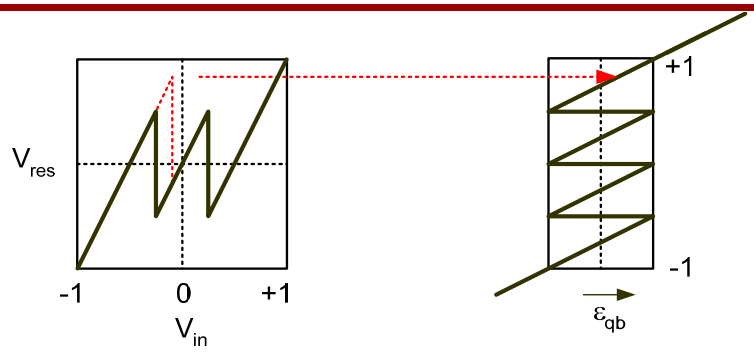
- Effective stage resolution is an integer
  - E.g. R = $\log_2 2 = 1 = B-1$
  - Digital hardware requires only a few adders, no need to implement fractional weights
- See e.g. [Mehr 2000]

Idea #3: G=2^B, Extended Backend Range



- No redundancy in stage with errors
- Extra decision levels in succeeding stage used to bring residue "back into the box"
- See e.g. [Opris 1998]

Variant of Idea #2: "1.5-bit stage"

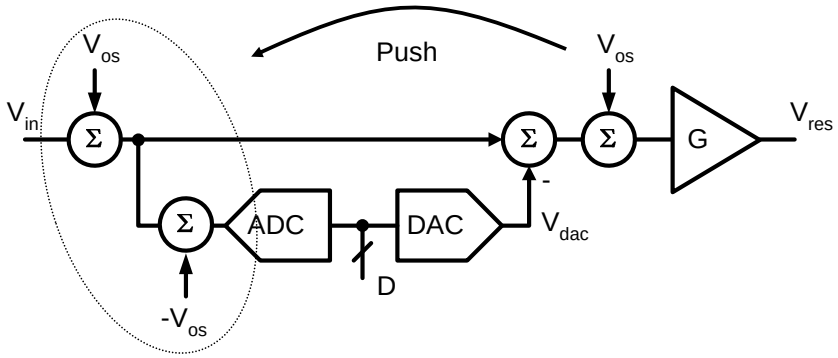


- Sub-ADC decision levels placed to minimize comparator count
- Can accommodate errors up to ±¼
- B = $\log_2(2+1)$ = 1.589 (sub-ADC resolution)
- R = $\log_2 2$ = 1 (effective stage resolution)
- See e.g. [Lewis 1992]

Summary on Sub-ADC Redundancy

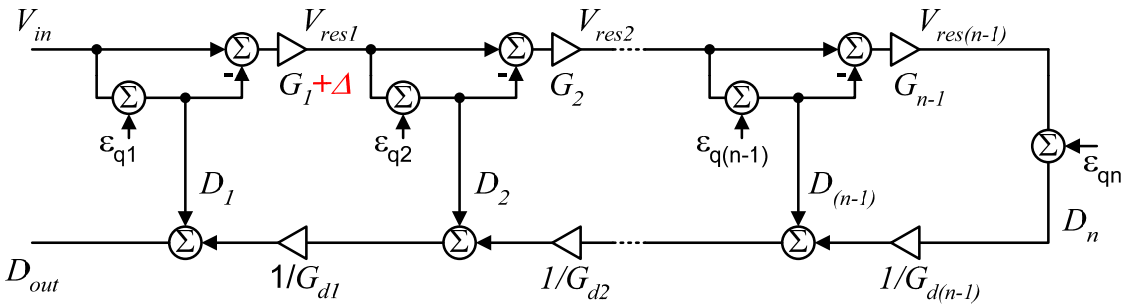
- We can tolerate sub-ADC errors as long as
  - The residue stays "inside the box", or
  - Another stage downstream returns the residue "into the box" before it reaches last quantizer
- This result applies to any stage in an n-stage pipeline
  - Can always decompose pipeline into single stage + backend ADC
- In literature, sub-ADC redundancy schemes are often called "digital correction" – a misnomer in my opinion
- There is no explicit error correction!
  - Sub-ADC errors are absorbed in the same way as their inherent quantization error
    - As long as there is no overranging…

Amplifier Offset



- Amplifier offset can be referred toward stage input and results in
  - Global offset
    - Usually no problem, unless "absolute ADC accuracy" is required
  - Sub-ADC offset
    - Easily accommodated through redundancy

Gain Errors



$$D_{out} = V_{in} + \varepsilon_{q1}\left(1 - \frac{G_1 + \Delta}{G_{d1}}\right) + \ldots + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_{dj}}$$

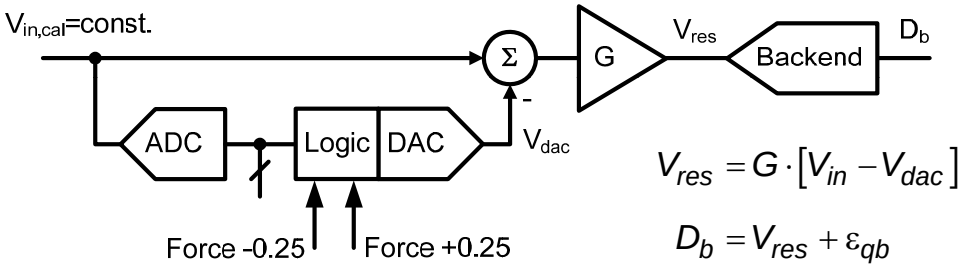
- Want to make $G_{d1} = G_1 + \Delta$

Digital Gain Calibration (1)

- Error in analog gain is not a problem as long as "digital gain term" is adjusted appropriately
- Problem
  - Need to measure analog gain precisely
- Example
  - Digital calibration of a 1-bit first stage with 1-bit redundancy (R=1, B=2)

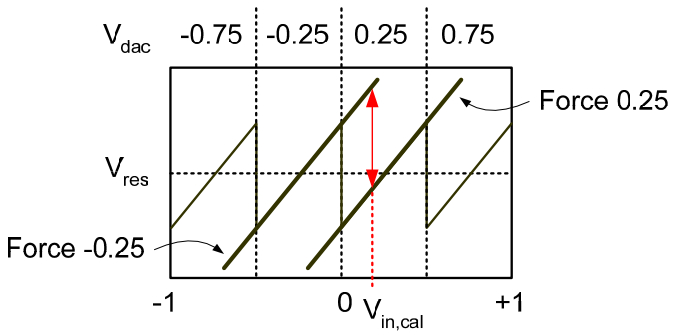
- Note
  - Even if all G_{dj} are perfectly adjusted to reflect the analog gains, the ADC will have non-zero DNL and INL, bounded by ±0.5LSB. This can be explained by the fact that the residue transitions may not correspond to integer multiples of the backend-LSB. This can cause non-uniformity in the ADC transfer function (DNL, INL) and also non-monotonicity (see [Markus, 2005]).
  - In case this cannot be tolerated
    - Add redundant bits to ADC backend (after combining all bits, final result can be truncated back)
    - Calibrate analog gain terms

Digital Gain Calibration (2)



$$V_{res} = G \cdot [V_{in} - V_{dac}]$$

$$D_b = V_{res} + \varepsilon_{qb}$$



Digital Gain Calibration (3)

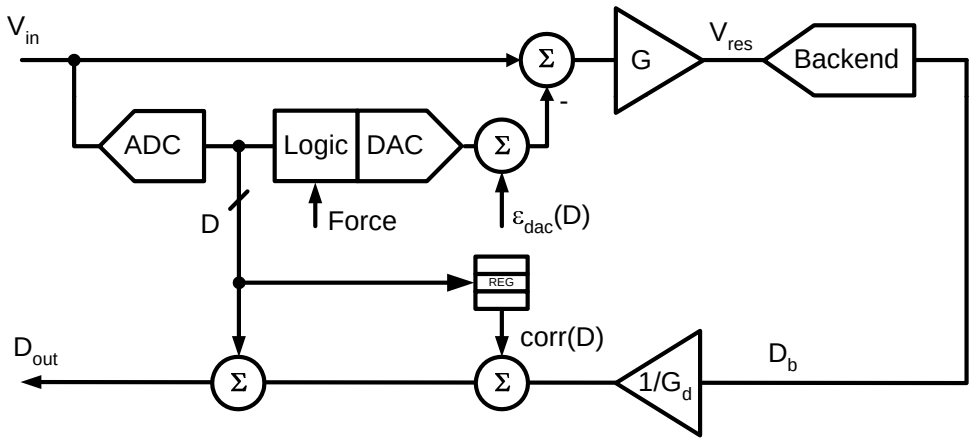
Step1: $$D_b^{(1)} = G \cdot \left[V_{in} + 0.25\right] + \varepsilon_{qb}^{(1)}$$

Step2: $$D_b^{(2)} = G \cdot \left[V_{in} - 0.25\right] + \varepsilon_{qb}^{(2)}$$

$$D_b^{(1)} - D_b^{(2)} = 0.5 \cdot G + \varepsilon_{qb}^{(1)} - \varepsilon_{qb}^{(2)}$$

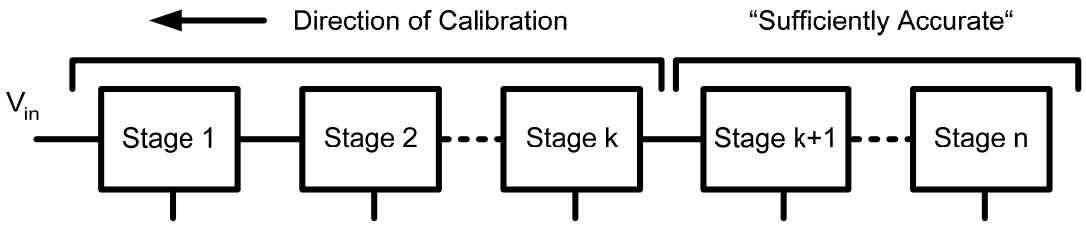
- Can minimize impact of quantization error using
  - Averaging (thermal noise dither)
  - Extra backend resolution

DAC Calibration



- Essentially same concept as gain calibration
  - Step through DAC codes and use backend to measure errors
- Store coefficients for each DAC transition in a look-up table

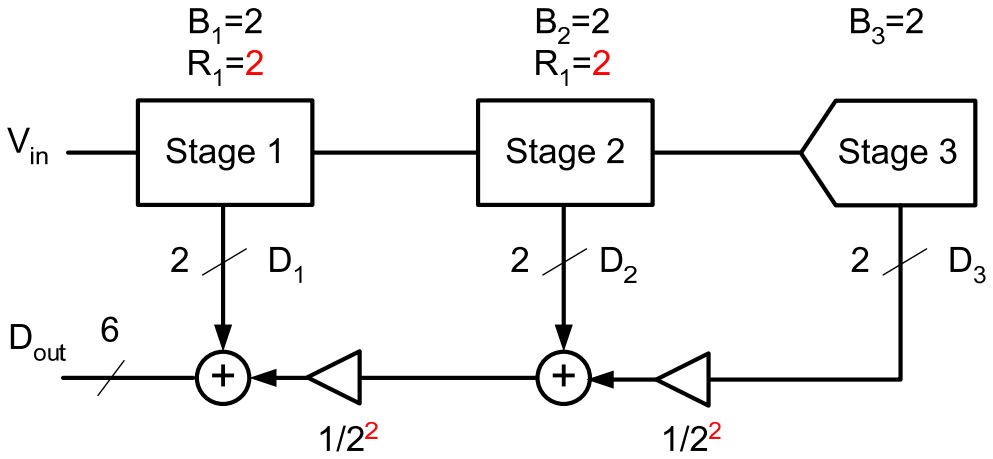
Recursive Stage Calibration



- First few stages have most stringent accuracy requirements
  - Errors of later stages are attenuated by aggregate gain
- Commonly used algorithm [Karanicolas 1993]
  - Take ADC offline
  - Measure least significant stage that needs calibration first
  - Move to next significant stage and continue toward stage 1

Combining the Bits (1)

- Example1: Three 2-bit stages, no redundancy

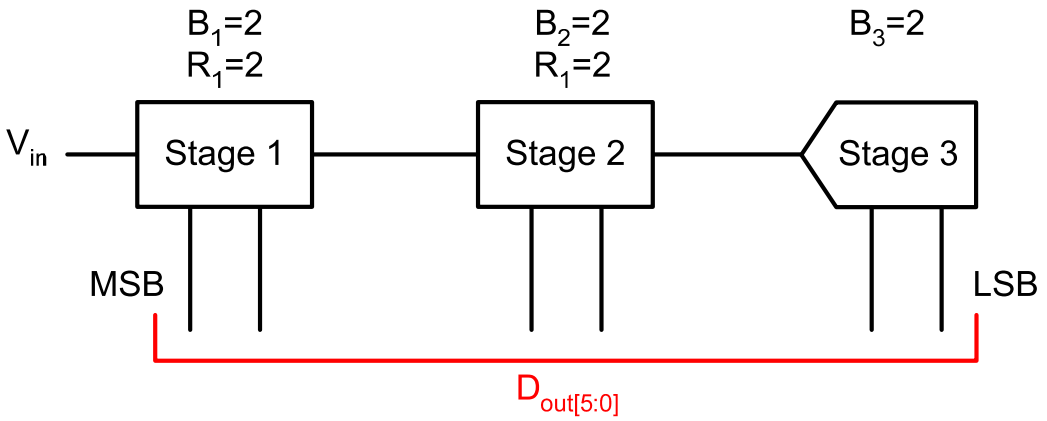


$$D_{out} = D_1 + \frac{1}{4}D_2 + \frac{1}{16}D_3$$

Combining the Bits (2)

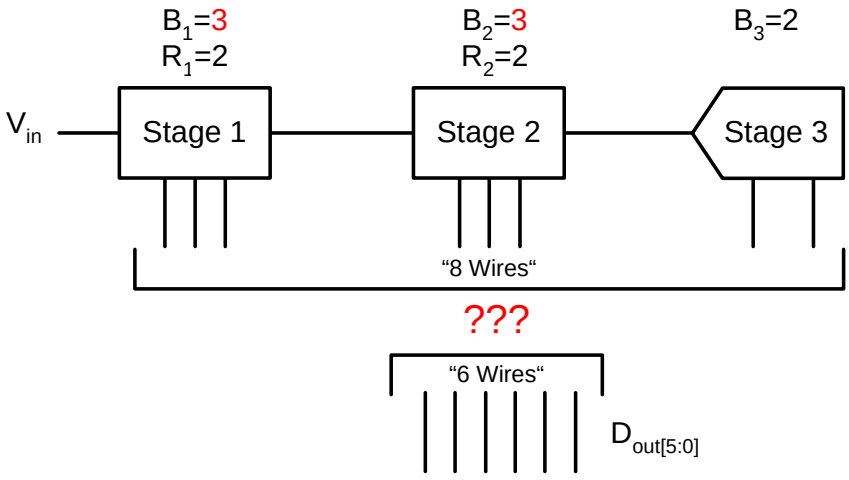
```
D1     XX
D2       XX
D3         XX
------------
Dout   DDDDDD
```

- Only bit shifts
- No arithmetic circuits needed



Combining the Bits (3)

- Example2: Three 2-bit stages, one bit redundancy in stages 1 and 2 (6-bit aggregate ADC resolution)

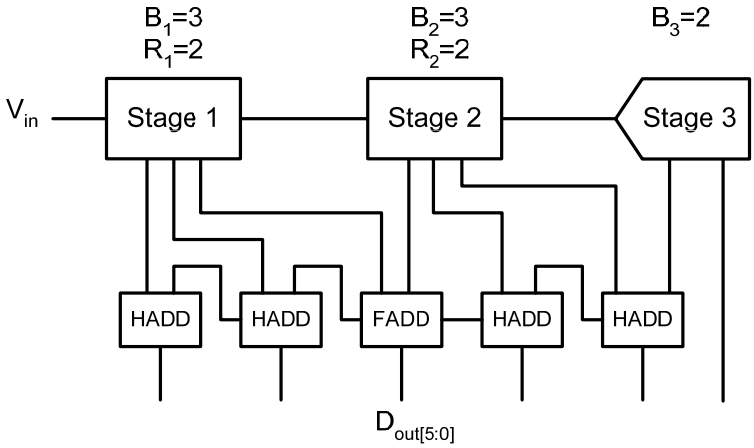


Combining the Bits (4)

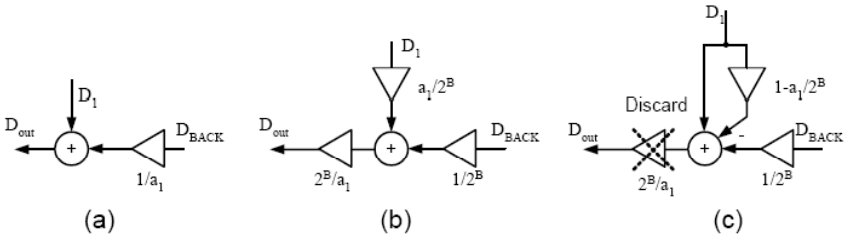
$$D_{out} = D_1 + \frac{1}{4}D_2 + \frac{1}{16}D_3$$

```
D1     XXX
D2       XXX
D3         XX
-------------
Dout   DDDDDD
```

- Bits overlap
- Need adders (Still, no good reason for calling this "digital correction"...)



Combining the Bits (5)

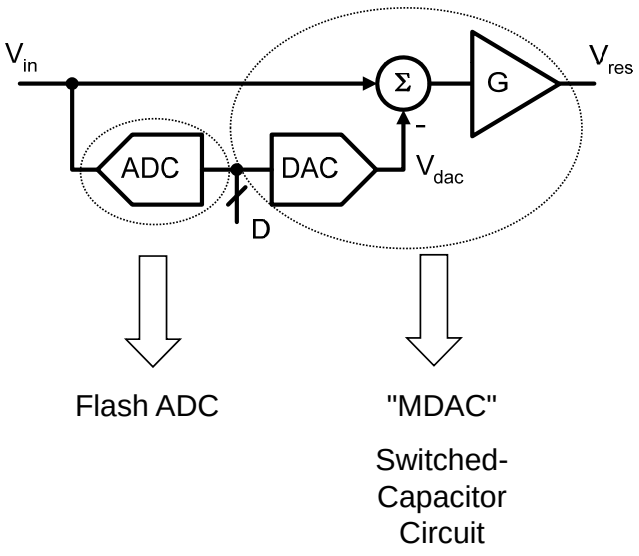


- For fractional weights (e.g. radix <2), there is no need to implement complex multipliers
- Can still use simple bit shifts; push actual multiplication into low-resolution output
  - E.g. a 1x10 bit multiplication needs only one adder…
- See e.g. [Karanicolas 1993]

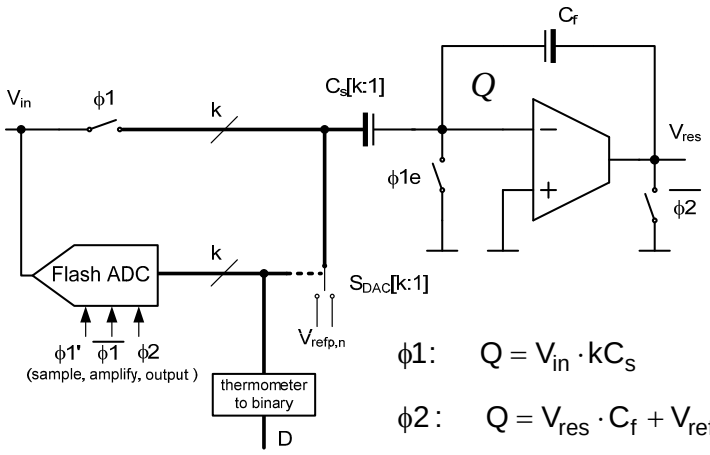
Outline

- Background
  - History and state-of the art performance
  - General idea of multi-step A/D conversion
- Pipeline ADC basics
  - Ideal block diagram and operation, impact of block nonidealities
- Ways to deal with nonidealities
  - Redundancy, calibration
- CMOS implementation details
  - Stage scaling, MDAC design
- Architectural options
  - OTA sharing, SHA-less front-end
- Research topics

Stage Implementation



Generic Circuit



$$\phi 1: \quad Q = V_{in} \cdot kC_s$$

$$\phi 2: \quad Q = V_{res} \cdot C_f + V_{refp} \cdot mC_s + V_{refn} \cdot (k-m)C_s$$

$$\therefore -V_{res} = \frac{kC_s}{C_f} V_{in} - \frac{mC_s}{C_f} V_{refp} + \frac{(k-m)C_s}{C_f} V_{refn}$$

$$= G \cdot (V_{in} - V_{dac})$$

Endless List of Design Parameters

- Stage resolution, stage scaling factor
- Stage redundancy
- Thermal noise/quantization noise ratio
- OTA architecture
  - OTA sharing?
- Switch topologies
- Comparator architecture
- Front-end SHA vs. SHA-less design
- Calibration approach (if needed)
- Time interleaving?
- Technology and technology options (e.g. capacitors)

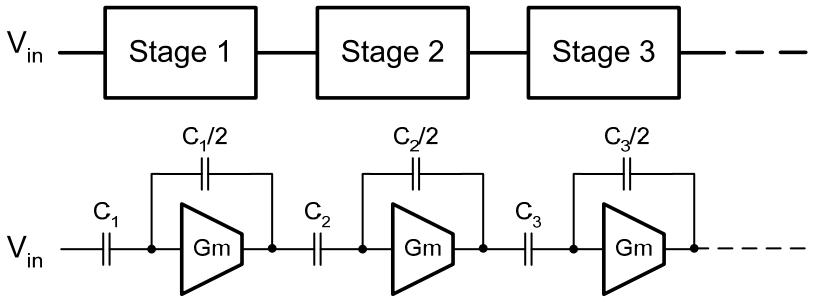
➢ A very complex optimization problem!

Thermal Noise Considerations

- Total input referred noise
  - Thermal noise + quantization noise
  - Costly to make thermal noise smaller than quantization noise
- Example: V_{FS}=1V, 10-bit ADC
  - $N_{quant}=LSB^2/12=(1V/2^{10})^2/12=(280\mu Vrms)^2$
  - Design for total input referred thermal noise ~280μVrms or larger, if SNR target allows
- Total input referred thermal noise is the sum of noise in all stages
  - How should we distribute the total thermal noise budget among the stages?
  - Let's look at an example…

Stage Scaling (1)

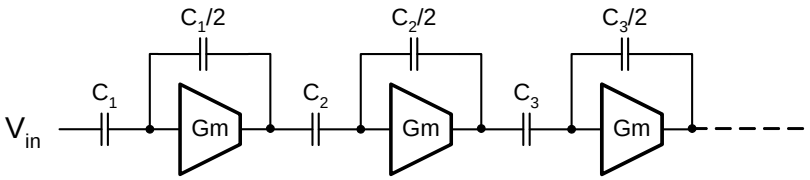
- Example: Pipeline using 1-bit (effective) stages (G=2)



- Total input referred noise power

$$N_{tot} \propto kT\left[\frac{1}{C_1}+\frac{1}{4C_2}+\frac{1}{16C_3}+...\right]$$

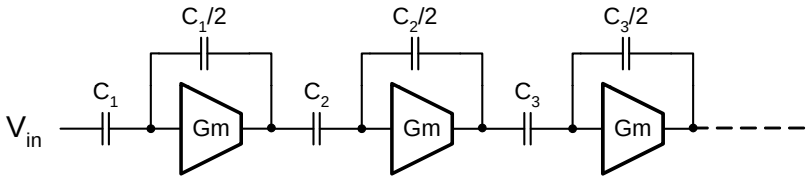
Stage Scaling (2)



$$N_{tot} \propto kT\left[\frac{1}{C_1}+\frac{1}{4C_2}+\frac{1}{16C_3}+...\right]$$

- If we make all caps the same size, backend stages contribute very little noise
- Wasteful, because Power ~ G_m ~ C

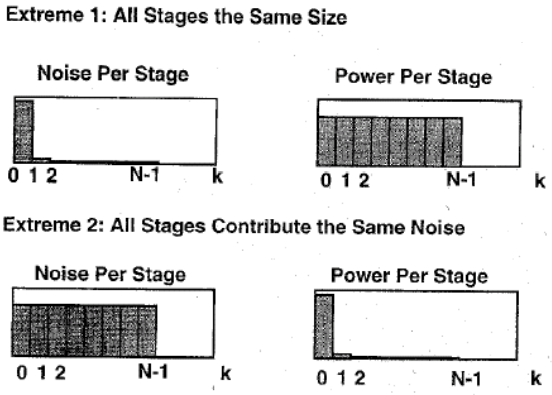
Stage Scaling (3)



$$N_{tot} \propto kT\left[\frac{1}{C_1}+\frac{1}{4C_2}+\frac{1}{16C_3}+...\right]$$

- How about scaling caps down by 2^2=4x per stage?
  - Same amount of noise from every stage
  - All stages contribute significant noise
  - Noise from first few stages must be reduced
  - Power ~ G_m ~ C goes up!

Stage Scaling (4)

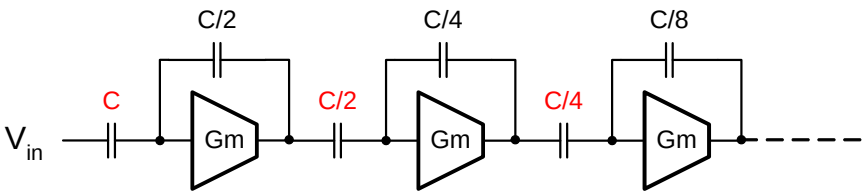


[Cline 1996]

- Optimum capacitior scaling lies approximately midway between these two extremes

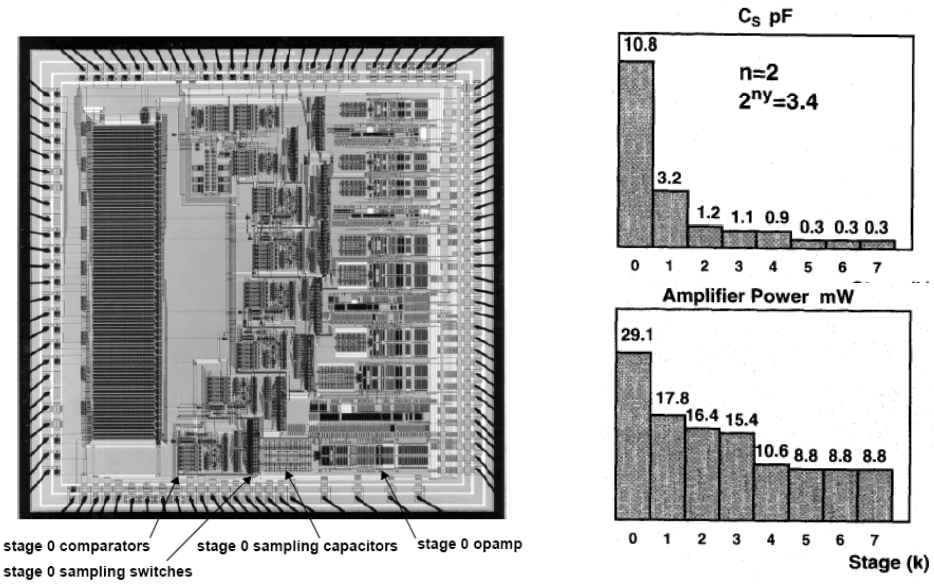
Practical Approach to Stage Scaling

- Start by assuming caps are scaled precisely by stage gain
  - E.g. for 1-bit effective stages:



- Refine using first pass circuit information & Excel spreadsheet
  - Use estimates of OTA power, parasitics, minimum feasible sampling capacitance etc.
- Or, buy a circuit optimization tool…

Stage Scaling Examples

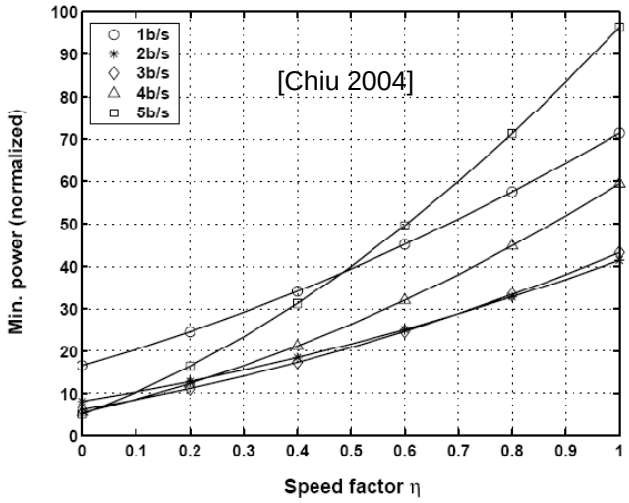


[Cline 1996]

How Many Bits Per Stage?

- Low per-stage resolution (e.g. 1-bit effective)
  - – Need many stages
  - + OTAs have small closed loop gain, large feedback factor
    - High speed
- High per-stage resolution (e.g. 3-bit effective)
  - + Fewer stages
  - – OTAs can be power hungry, especially at high speed
  - – Significant loading from flash-ADC
- Qualitative conclusion
  - – Use low per-stage resolution for very high speed designs
  - – Try higher resolution stages when power efficiency is most important constraint

Power Tradeoff is Fairly Flat!



η = parasitic cap at output/total sampling cap in each stage (junctions, wires, switches, …)

- ADC power varies by only ~2x across different stage resolutions!

Examples

Reference	[Yoshioka, 2007]	[Jeon, 2007]	[Loloee 2002]	[Bogner 2006]
Technology	90nm	90nm	0.18um	0.13um
Bits	10	10	12	14
Bits/Stage	1-1-1-1-1-1-1-3	2-2-2-4	1-1-1-1-1-1-1-1-1-1-2	3-3-2-2-4
SNDR [dB]	~56	~54	~65	~64
Speed [MS/s]	80	30	80	100
Power [mW]	13.3	4.7	260	224
mW/MS/s	0.17	0.16	3.25	2.24

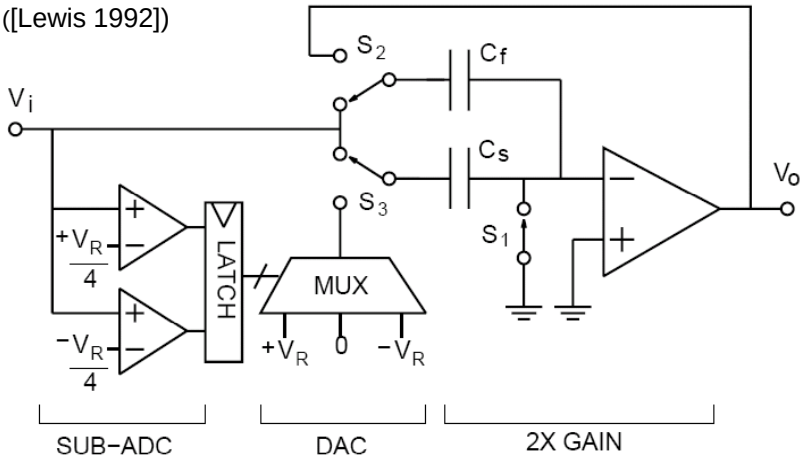
- Low power is possible for a wide range of architectures!

Re-Cap

- Choosing the "optimum" per-stage resolution and stage scaling scheme is a non-trivial task
  - But – optima are shallow!
- Quality of transistor level design and optimization is at least as important (if not more important than) architectural optimization…
- Next, look at circuit design details
  - Assume we're trying to build a 10-bit pipeline
    - Recent technology, feature size ~0.18μm or smaller
    - Moderate to high-speed ~100MS/s
    - 1-bit effective/stage, using "1.5-bit" stage topology
    - Dedicated front-end SHA

1.5-Bit Stage Implementation

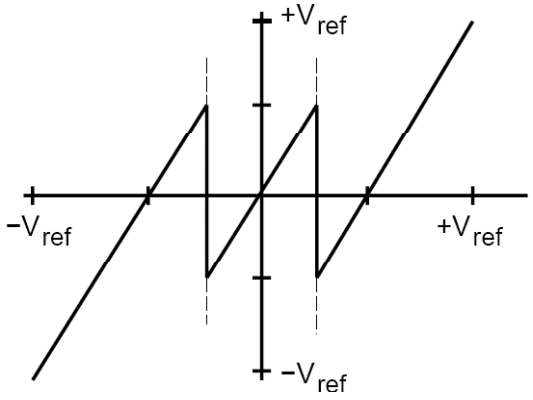
[Abo 1999] ([Lewis 1992])



- C_f is used as sampling cap during acquisition phase, as feedback cap in redistribution phase
  - Helps improve feedback factor (max. 1/3 → max. 1/2)

Residue Plot

$$V_o = \begin{cases} \left(1 + \frac{C_s}{C_f}\right) V_i - \frac{C_s}{C_f} V_{ref} & \text{if } V_i > V_{ref}/4 \\ \left(1 + \frac{C_s}{C_f}\right) V_i & \text{if } -V_{ref}/4 \leq V_i \leq +V_{ref}/4 \\ \left(1 + \frac{C_s}{C_f}\right) V_i + \frac{C_s}{C_f} V_{ref} & \text{if } V_i < -V_{ref}/4 \end{cases}$$

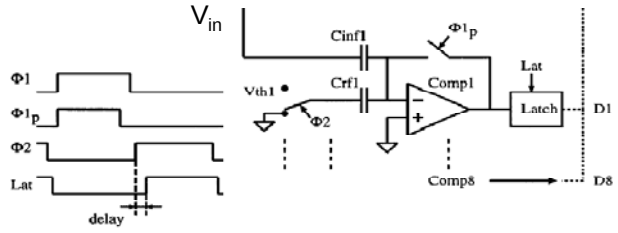


[Abo 1999]

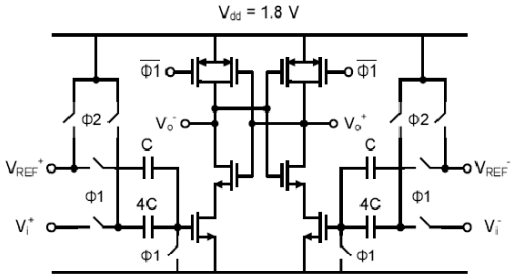
Comparators

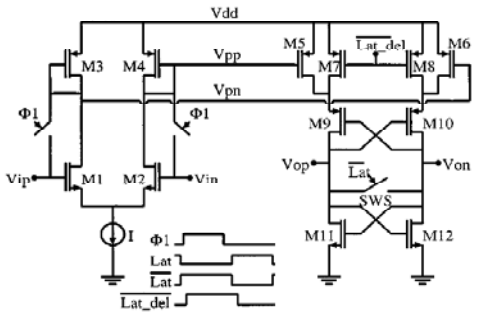
- Can tolerate large offsets and large noise with appropriate redundancy
- Consume negligible power in a good design
  - 50-100μW or less per comparator
- Lots of implementation options
  - Resistive/capacitive reference generation
  - Different pre-amp/latch topologies
  - …

Comparator Examples



[Chiu 2004]



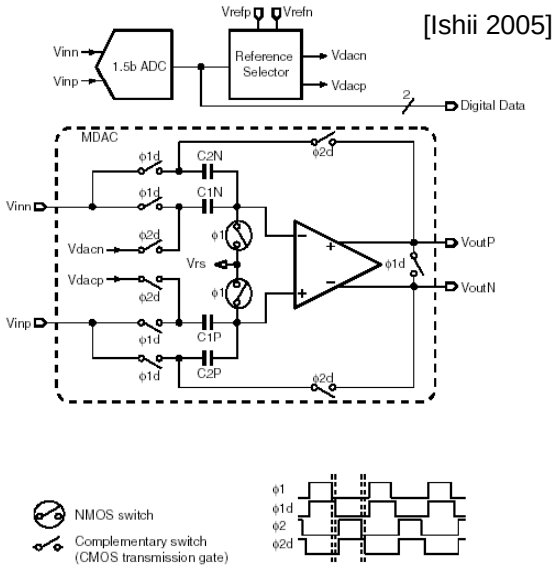


[Mehr 2000]

OTA Design Considerations

- Static amplifier error = 1/(DC Loop Gain)
  - E.g. for 0.1% accuracy in first stage of 10-bit ADC, need loop gain > 60dB
- Dynamic settling error
  - Typically want to settle outputs to ~1/8 LSB accuracy within 1/2 clock cycle
- Thermal noise
  - Size capacitors to satisfy kT/C noise requirement
- Start by picking an OTA topology that will deliver sufficient gain
  - Or think about ways to compensate finite gain error…
- General references on OTA design
  - [Boser 2005], [Murmann, EE315A]

Switches



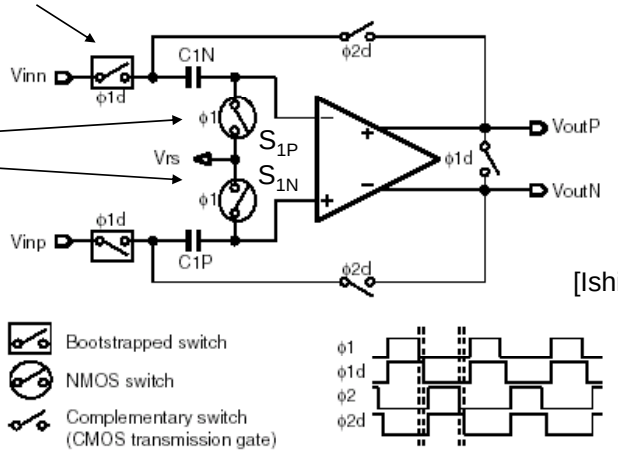
[Ishii 2005]

- Make switch RC ~ 10 times faster than OTA
  - Avoids speed degradation
  - Minimizes switch noise contribution
    - See e.g. [Schreier 2005]
  - Avoids stability issues due to poles in feedback network
- Three choices for switches
  - Single N or P device
  - Transmission gate
  - Bootstrapped NMOS
    - For high swing nodes that require constant R_{on}

Front-End SHA

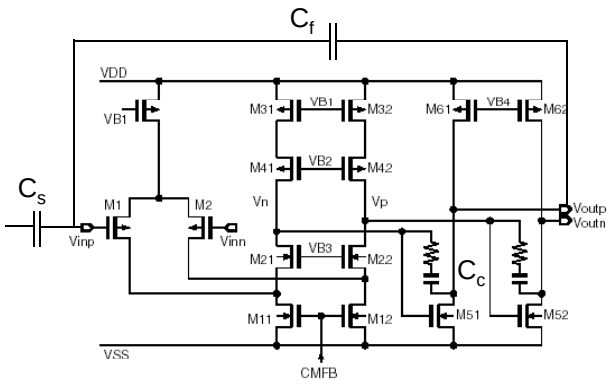
Need constant R_{ON} here to minimize signal dependent charge injection from S_{1N}, S_{1P}

Minimize Jitter!



[Ishii 2005]

Total Integrated OTA Noise (1)



$$N_1 = 1 + \frac{g_{m11} + g_{m31}}{g_{m1}} \cong 2...4$$

$$N_2 = 1 + \frac{g_{m61}}{g_{m51}} \cong 2$$

$$\overline{V_{od}^2} = \underbrace{2\frac{1}{\beta}\cdot\gamma\cdot N_1\frac{kT}{C_c}}_{\text{Stage 1}} + \underbrace{2(\gamma\cdot N_2+1)\frac{kT}{C_{Ltot}}}_{\text{Stage 2}}$$

ignore in first cut design

$$\beta = \frac{C_f}{C_f + C_s + C_{gs1}}$$

$$C_{Ltot} = C_L + (1-\beta)C_f + C_{parasitic}$$

Total Integrated OTA Noise (2)

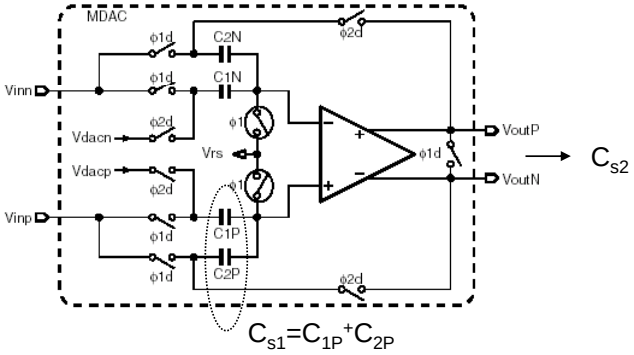
- Assuming γ=1, N_1=N_2=2

$$\overline{V_{od}^2} = 4\frac{1}{\beta}\cdot\frac{kT}{C_c} + 6\frac{kT}{C_{Ltot}}$$

- OTA noise partitioning problem
  - How should we split noise between stage1 and stage2 terms?
- In this design example we'll use a 2/3, 1/3 split
  - This is yet another design/optimization parameter
- With this assumption, we have

$$\overline{V_{od}^2} = 18\frac{kT}{C_{Ltot}} \qquad C_c = \frac{C_{Ltot}}{3\beta}$$

Stage 1 Noise



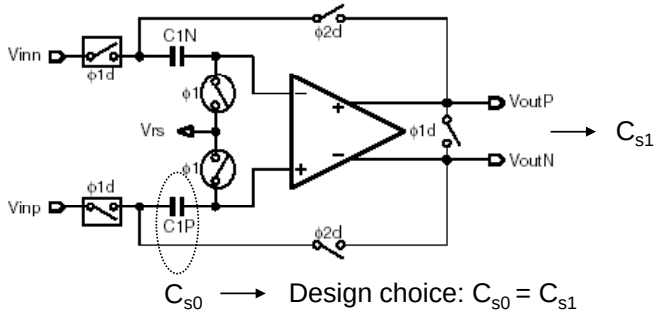
$$\beta = \frac{C_{s1}/2}{C_{s1}+C_{gs1}} \cong \frac{1}{3}$$

$$\overline{V_{od,1}^2} = 18\frac{kT}{C_{s2}+C_{s1}/3}$$

$$C_{Ltot} = C_{s2} + \left(1-\frac{1}{3}\right)\frac{C_{s1}}{2}$$

$$\overline{V_{id,1}^2} = \frac{18}{2^2}\frac{kT}{C_{s2}+C_{s1}/3}$$

SHA Noise



$$\beta = \frac{C_{s0}}{C_{s0}+C_{gs1}} \cong \frac{1}{2}$$

$$C_{Ltot} = C_{s1} + \left(1-\frac{1}{2}\right)\frac{C_{s0}}{2}$$

From sample phase (ϕ1)

$$\overline{V_{od,0}^2} = \overline{V_{id,0}^2} = 18\frac{kT}{C_{s1}+C_{s0}/4} + \frac{kT}{C_{s0}} \cong 16\frac{kT}{C_{s1}}$$

Noise Budgeting

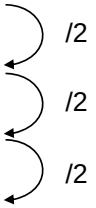
- Total input referred noise budget, assuming $V_{FS,diff}$=1V
  - $N_{thermal} = N_{quant}$=LSB^2/12=(1V/2^{10})^2/12=(280μVrms)^2
- Reasonable "first cut" partitioning of input referred noise
  - SHA → 1/2
  - Stage 1 → 1/4
  - All remaining stages → 1/4

$$\overline{V_{id,0}^2} = 16\frac{kT}{C_{s1}} = \frac{1}{2}\left(280\mu Vrms\right)^2 \Rightarrow C_{s1} = 1.66pF$$

$$\overline{V_{id,1}^2} = \frac{9}{2}\frac{kT}{C_{s2} + C_{s1}/3} = \frac{1}{4}\left(280\mu Vrms\right)^2 \Rightarrow C_{s2} = 0.38pF$$

Capacitor Sizes

C_{s0}	1.66pF
C_{s1}	1.66pF
C_{s2}	0.38pF
C_{s3}	190fF
C_{s4}	85fF
C_{s5}	42.fF (minimum)
…	…
C_{s10}	42.fF (minimum)



- Now refine these numbers using simulation and Excel spreadsheet
  - Iterate over assumptions/design choices to optimize design

Reality Check

[Honda 2007]

STAGE	C_s [pF]	Power [mW]
S/H	2.0	3.5
STAGE1	1.0	3.0
STAGE2	0.5	2.0
STAGE3	0.3	2.0
STAGE4	0.3	1.8
STAGE5	0.16	1.8
STAGE6	0.16	1.5
STAGE7-10	0.1	1.5
Others [Bias circuits, Clock gen.]		5.0
Total static power		26.6

Technology	90nm digital CMOS
Supply voltage	1.0 V
Resolution	10 bits
Sampling rate	100 MSample/s
Full scale analog input	0.8 V_{pp}
Maximum DNL	-0.7/+0.3 LSB
Maximum INL	-0.6/+0.7 LSB
SNDR ($F_{in} \cong 10MHz$)	55.3dB
SFDR ($F_{in} \cong 10MHz$)	71.5dB
Total power consumption	33mW
Packaging	Chip-on-board
Active area	1.3mm×3.1mm

- Not too far off from a practical design…

Kawahito's Design Charts (1)

(1-bit effective in each stage)

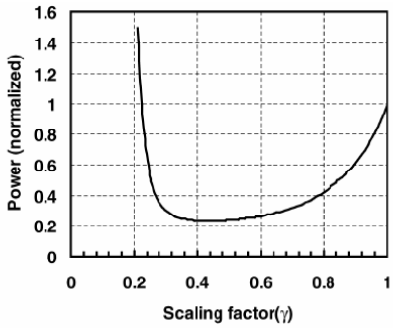


Fig.3. Power versus scaling factor of capacitors (10b ADC).

[Kawahito 2006]

(using single-stage OTA model)

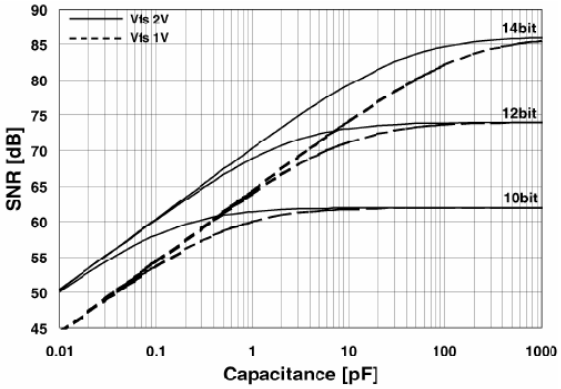


Fig. 9 SNR versus the sampling capacitor of the first stage of the MDAC (n=2, a=1, b=0.5, ζ=2, γ=0.5) .

Kawahito's Design Charts (2)

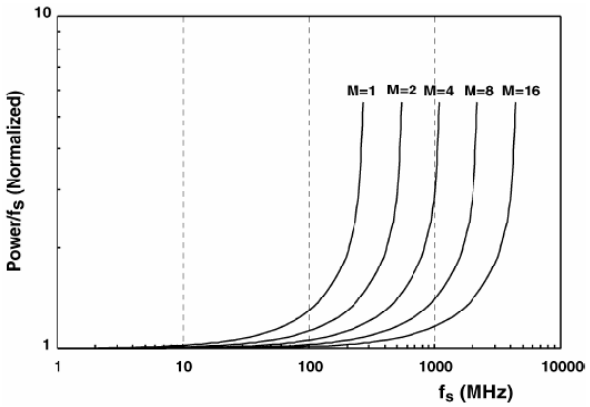


Fig. 13. Power/f_s versus f_s in parallel pipeline ADCs (# of channels (M) is 1, 2, 4, 8 and 16).

- Consider time-interleaving at high f_s

[Kawahito 2006]

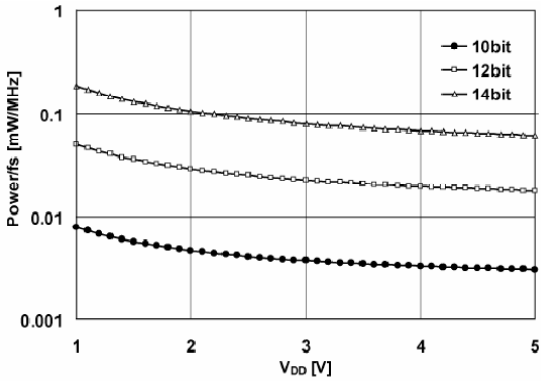


Fig. 14 Power normalized by sampling frequency of ideal pipeline ADCs versus V_{DD}.

- In theory, plenty of room for power improvement…

Selected References (1)

- General
  - S. Kawahito, "Low-Power Design of Pipeline A/D Converters," Proc. CICC, pp. 505-512, Sep. 2006.
- Redundancy & Calibration
  - S. H. Lewis et al., "A 10-b 20-Msample/s analog-to-digital converter," IEEE JSSC, pp. 351-358, Mar. 1992.
  - A. N. Karanicolas et al. "A 15-b 1-Msample/s digitally self-calibrated pipeline ADC," IEEE J. Of Solid-State Circuits, pp. 1207-1215, Dec. 1993.
  - E. G. Soenen et al., "An architecture and an algorithm for fully digital correction of monolithic pipelined ADCs," IEEE TCAS II, pp. 143-153, Mar. 1995.
  - I. E. Opris et al., "A single-ended 12-bit 20 MSample/s self-calibrating pipeline A/D converter," IEEE JSSC, pp. 1898-1903, Dec. 1998.
  - L. Singer et al., "A 12 b 65 MSample/s CMOS ADC with 82 dB SFDR at 120 MHz," ISSCC Dig. Techn. Papers, pp. 38-39, Feb. 2000.
  - I. Mehr et al., "A 55-mW, 10-bit, 40-Msample/s Nyquist-rate CMOS ADC," IEEE JSSC, pp. 318-325, Mar. 2000.
  - J. Ming et al., "An 8-bit 80-Msample/s pipelined analog-to-digital converter with background calibration," IEEE JSSC, pp. 1489-1497, Oct. 2001.
  - S.-Y. Chuang et al., "A Digitally Self-Calibrating 14-bit 10-MHz CMOS Pipelined A/D Converter," IEEE JSSC, pp. 674-683, Jun. 2002.
  - J. Markus et al., "On the monotonicity and linearity of ideal radix-based A/D converters," IEEE Trans. Instr. and Measurement, pp. 2454-2457, Dec. 2005.

Selected References (2)

- Implementation
  - T. Cho, "Low-Power Low-Voltage Analog-to-Digital Conversion Techniques using Pipelined Architecures, PhD Dissertation, UC Berkeley, 1995, http://kabuki.eecs.berkeley.edu/~tcho/Thesis1.pdf.
  - L. A. Singer at al., "A 14-bit 10-MHz calibration-free CMOS pipelined A/D converter," VLSI Circuit Symposium, pp. 94-95, Jun. 1996.
  - A. Abo, "Design for Reliability of Low-voltage, Switched-capacitor Circuits," PhD Dissertation, UC Berkeley, 1999, http://kabuki.eecs.berkeley.edu/~abo/abothesis.pdf.
  - D. Kelly et al., "A 3V 340mW 14b 75MSPS CMOS ADC with 85dB SFDR at Nyquist," ISSCC Dig. Techn. Papers, pp. 134-135, Feb. 2001.
  - A. Loloee, et. al, “A 12b 80-MSs Pipelined ADC Core with 190 mW Consumption from 3 V in 0.18-um, Digital CMOS”, Proc. ESSCIRC, pp. 467-469, 2002
  - B.-M. Min et al., "A 69-mW 10-bit 80-MSample/s Pipelined CMOS ADC," IEEE JSSC, pp. 2031-2039, Dec. 2003.
  - Y. Chiu, et al., "A 14-b 12-MS/s CMOS pipeline ADC with over 100-dB SFDR,” IEEE JSSC, pp. 2139-2151, Dec. 2004.
  - S. Limotyrakis et al., "A 150-MS/s 8-b 71-mW CMOS time-interleaved ADC," IEEE JSSC, pp. 1057-1067, May 2005.
  - T. N. Andersen et al., "A Cost-Efficient High-Speed 12-bit Pipeline ADC in 0.18-um Digital CMOS," IEEE JSSC, pp. 1506-1513, Jul 2005.

Selected References (3)

  - P. Bogner et al., "A 14b 100MS/s digitally self-calibrated pipelined ADC in 0.13um CMOS," ISSCC Dig. Techn. Papers, pp. 832-833, Feb. 2006.
  - D. Kurose et al., "55-mW 200-MSPS 10-bit Pipeline ADCs for Wireless Receivers," IEEE JSSC, pp. 1589-1595, Jul. 2006.
  - S. Bardsley et al., "A 100-dB SFDR 80-MSPS 14-Bit 0.35um BiCMOS Pipeline ADC," IEEE JSSC, pp. 2144-2153, Sep. 2006.
  - A. M. A. Ali et al, "A 14-bit 125 MS/s IF/RF Sampling Pipelined ADC With 100 dB SFDR and 50 fs Jitter," IEEE JSSC, pp. 1846-1855, Aug. 2006.
  - S. K. Gupta, "A 1-GS/s 11-bit ADC With 55-dB SNDR, 250-mW Power Realized by a High Bandwidth Scalable Time-Interleaved Architecture," IEEE JSSC, 2650-2657, Dec. 2006.
  - M. Yoshioka et al., "A 0.8V 10b 80MS/s 6.5mW Pipelined ADC with Regulated Overdrive Voltage Biasing," ISSCC Dig. Techn. Papers, pp. 452-453, Feb. 2007.
  - Y.-D. Jeon et al., "A 4.7mW 0.32mm^2 10b 30MS/s Pipelined ADC Without a Front-End S/H in 90nm CMOS," ISSCC Dig. Techn. Papers, pp. 456-457, Feb. 2007.
  - K.-H. Lee et al., "Calibration-free 14b 70MS/s 0.13um CMOS pipeline A/D converters based on highmatching 3D symmetric capacitors," Electronics Letters, pp. 35-36, Mar. 15, 2007.
  - K. Honda et al., "A Low-Power Low-Voltage 10-bit 100-MSample/s Pipeline A/D Converter Using Capacitance Coupling Techniques," IEEE JSSC, pp. 757-765, Apr. 2007.

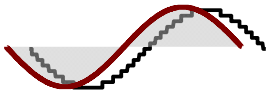
Selected References (4)

- Per-Stage Resolution and Stage Scaling
  - D. W. Cline, "Noise, Speed, and Power Tradeoffs in Pipelined Analog to Digital Converters", PhD Dissertation, UC Berkeley, November, 1995, http://kabuki.eecs.berkeley.edu/~cline/thesis/thesis.pdf.
  - D. W. Cline et al., "A power optimized 13-b 5 MSamples/s pipelined analog-to-digital converter in 1.2um CMOS," IEEE JSSC, Mar. 1996
  - Y. Chiu, "High-Performance Pipeline A/D Converter Design in Deep-Submicron CMOS," PhD Dissertation, UC Berkeley, 2004.
  - H. Ishii et al., "A 1.0 V 40mW 10b 100MS/s pipeline ADC in 90nm CMOS," Proc. CICC, pp. 395-398, Sep. 2005.
- OTA Design, Noise
  - B. E. Boser, "Analog Circuit Design with Submicron Transistors," Presentation at IEEE Santa Clara Valley, May 19, 2005, http://www.ewh.ieee.org/r6/scv/ssc/May1905.htm
  - B. Murmann, EE315A Course Material, http://ccnet.stanford.edu/cgi-bin/handouts.cgi?cc=ee315a
  - R. Schreier et al., "Design-oriented estimation of thermal noise in switched-capacitor circuits," IEEE TCAS I, pp. 2358-2368, Nov. 2005.

Selected References (5)

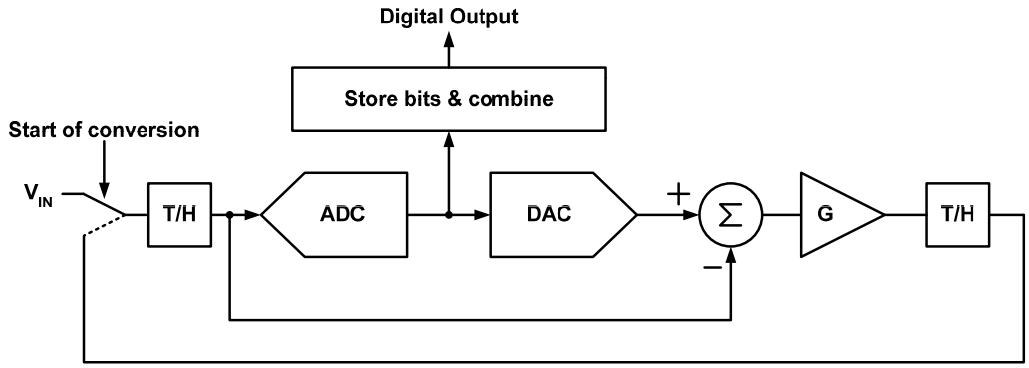
- Capacitor Matching Data
  - C. H. Diaz et al., "CMOS technology for MS/RF SoC," IEEE Trans. Electron Devices, pp. 557-566, Mar. 2003.
  - A. Verma et al., "Frequency-Based Measurement of Mismatches Between Small Capacitors," Proc. CICC, pp. 481-484, Sep. 2006.
- Reference Generator
  - T. L. Brooks et al., "A low-power differential CMOS bandgap reference," ISSCC Dig. Techn. Papers, pp. 248-249, Feb. 1994.
- Research
  - B. Murmann et al., "A 12-bit 75-MS/s Pipelined ADC using Open-Loop Residue Amplification," IEEE JSSC, pp. 2040-2050, Dec. 2003.
  - J. Fiorenza et al., "Comparator-Based Switched-Capacitor Circuits for Scaled CMOS Technologies, " IEEE JSSC, pp. 2658-2668, Dec. 2006.
  - E. Iroaga et al. "A 12b, 75MS/s Pipelined ADC Using Incomplete Settling," IEEE JSSC, pp. 748-756, Apr. 2007.
  - J. Hu, N. Dolev and B. Murmann, "A 9.4-bit, 50-MS/s, 1.44-mW Pipelined ADC using Dynamic Residue Amplification," VLSI Circuits Symposium, June 2008.

Bit-at-a-Time ADCs
Time Interleaving



Katelijn Vleugels
Stanford University

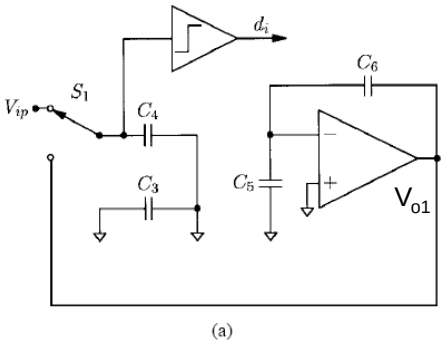
Cyclic ADC



- Essentially same as pipeline, but a single stage is used in a cyclic fashion for all operations
- Need many clock cycles per conversion

Implementation Example

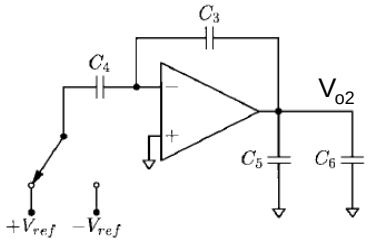
$\phi 1$



(a)

$$V_{o1} = \left(1 + \frac{C_5}{C_6}\right) V_{o2}$$

$\phi 2$



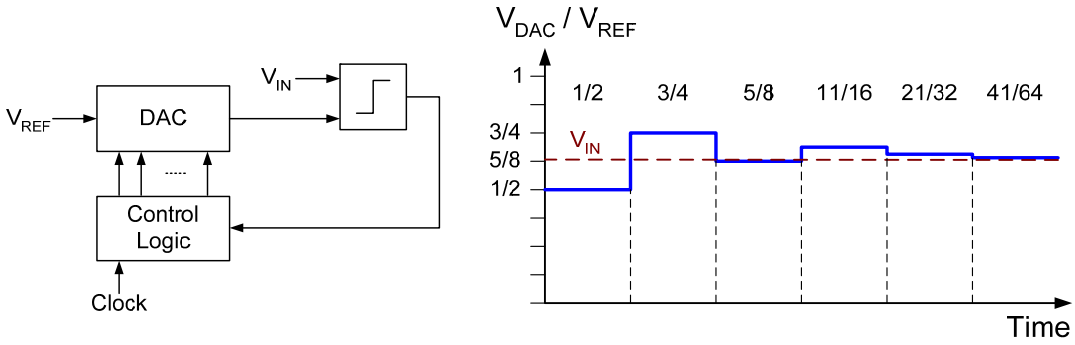
[Erdogan et al. JSSC 12/99]

$$V_{o2} = \left(V_{ip} \pm V_{ref}\right) \frac{C_4}{C_3}$$

Discussion

- Advantages
  - Area efficient
    - Typically only one or two switched capacitor stages plus comparator
  - Easy to calibrate
    - Need to measure only one coefficient (capacitor ratio)
- Disadvantages
  - Slow
    - Need many clock cycles for a single conversion
  - Sub-optimal power efficiency
    - Cannot scale stages like in a pipeline ADC
    - Noise and accuracy requirements decrease from MSB to LSB cycle, but invested circuit energy per cycle is (usually) constant

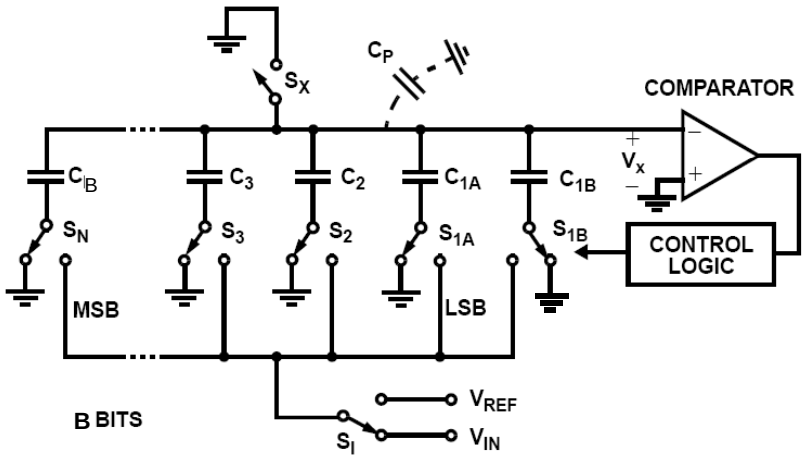
Successive Approximation Register ADC



- Binary search over DAC output
- High accuracy achievable (16+ Bits)
  - Relies on highly accurate comparator
- Moderate speed (1+ MHz)

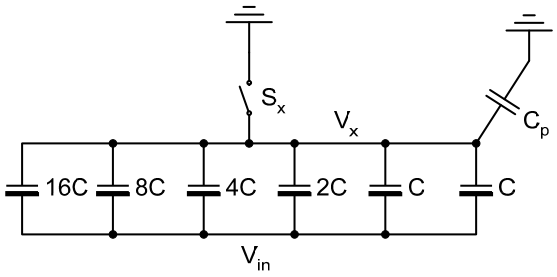
Implementation

- See e.g. [McCreary, JSSC 12/1975]



$C_{1A} = C_{1B} = C \qquad C_2 = 2C \qquad C_3 = 4C \qquad C_B = 2^{B-1}C$

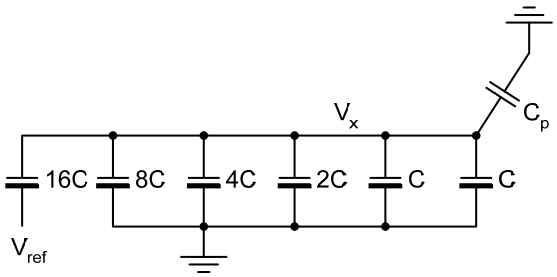
Sampling Phase (5-bit Example)



- Sample: S_x closes, S_{1A}, S_{1B}, S_2,…, S_5 connected to V_{in}
- Total charge at node V_x after opening S_x

$$Q = -V_{in} \cdot 32C = -V_{in} \cdot C_{total}$$

Bit5 Test (MSB)

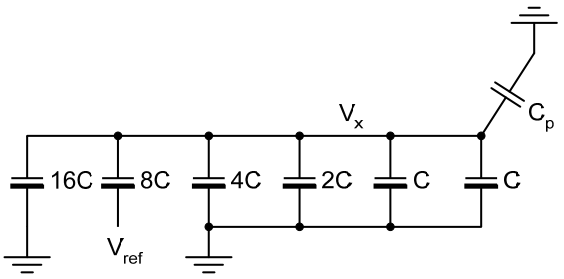


$$Q = -V_{in} \cdot C_{total} = (V_x - V_{ref}) \cdot 16C + V_x \cdot (16C + C_p)$$

$$\therefore V_x = \left(\frac{1}{2}V_{ref} - V_{in}\right) \cdot \frac{C_{total}}{C_{total} + C_p}$$

- $V_x<0 \Rightarrow V_{in}>0.5V_{ref} \Rightarrow$ Bit5=1
- $V_x>0 \Rightarrow V_{in}<0.5V_{ref} \Rightarrow$ Bit5=0

Bit4 Test (Assuming bit5=0)



$$Q = -V_{in} \cdot C_{total} = (V_x - V_{ref}) \cdot 8C + V_x \cdot (24C + C_p)$$

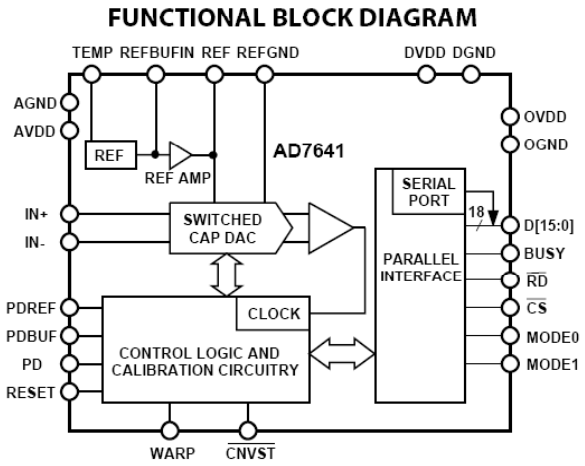
$$\therefore V_x = \left(\frac{1}{4}V_{ref} - V_{in}\right) \cdot \frac{C_{total}}{C_{total} + C_p}$$

- $V_x<0 \Rightarrow V_{in}>0.25V_{ref} \Rightarrow$ Bit4=1
- $V_x>0 \Rightarrow V_{in}<0.25V_{ref} \Rightarrow$ Bit4=0

Limitations

- Conversion rate typically limited by finite bandwidth of RC network during sampling and bit-tests
- For high resolution, the binary weighted capacitor array can become quite large
  - E.g. 16-bit resolution, C_{total}~100pF for reasonable kT/C noise contribution
- If matching is an issue, an even larger value may be needed
  - E.g. if matching dictates C_{min}=10fF, then $2^{16}C_{min}$=655pF
- Commonly used techniques
  - Implement "two-stage" or "multi-stage" capacitor network to reduce array size [Yee, JSSC 8/79]
  - Calibrate capacitor array to obtain precision beyond raw technology matching [Lee, JSSC 12/84]

High Performance Example

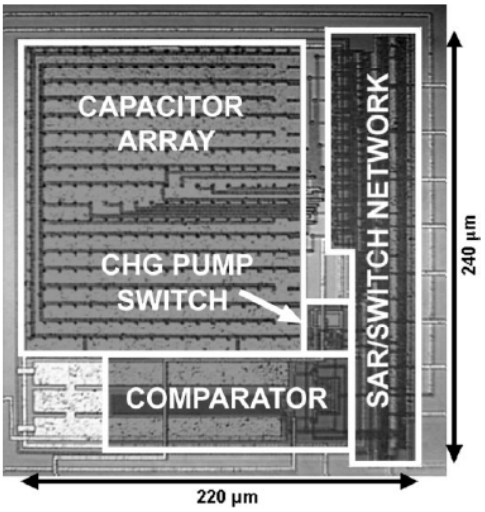


FEATURES

Throughput:
- **2 MSPS (Warp mode)**
- **1.5 MSPS (Normal mode)**

18-bit resolution with no missing codes
2.048V internal low drift refernce
INL: ±2 LSB typical
S/(N+D): 93 dB typical @ 20 kHz
THD: −115 dB typical @ 20 kHz
Differential input range: ±V_{REF} (V_{REF} up to 2.5 V)
No pipeline delay (SAR architecture)
Parallel (18-, 16-, or 8-bit bus)
Serial 5 V/3.3 V/2.5 V interface
SPI®/QSPI™/MICROWIRE™/DSP compatible
Single 2.5 V supply operation
Power dissipation: 65 mW typical @ 2 MSPS

Low Power Example

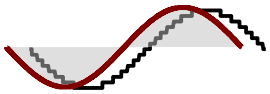


SUMMARY OF ADC PERFORMANCE

Performance Metric	Value
Voltage supply	1 V (nominal)
Input range	Rail-to rail
Sampling rate	100 kHz
Unit capacitance	12 fF
DNL	< ± 0.5 LSB typical
INL	< ± 0.5 LSB typical
ENOB (1V)	7.9 (DC), 7.0 (4.61 kHz)
Power dissipation (1V)	3.1 µW
Energy per sample (1V)	31 pJ
Standby power (1V)	70 pW
Die area (active)	0.053 mm^2
Process	0.25 µm CMOS (2P5M)

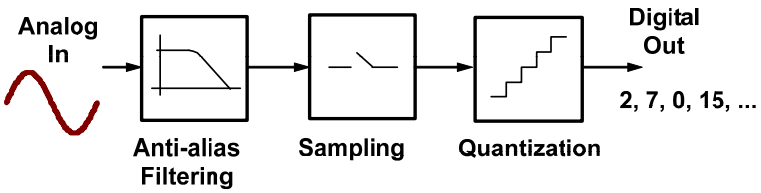
M.D. Scott, B.E. Boser, K.S.J. Pister, "An ultralow-energy ADC for Smart Dust," IEEE J. Solid-State Circuits, pp. 1123 -1129, July 2003.

Oversampling A/D Conversion



Katelijn Vleugels
Stanford University

Recap

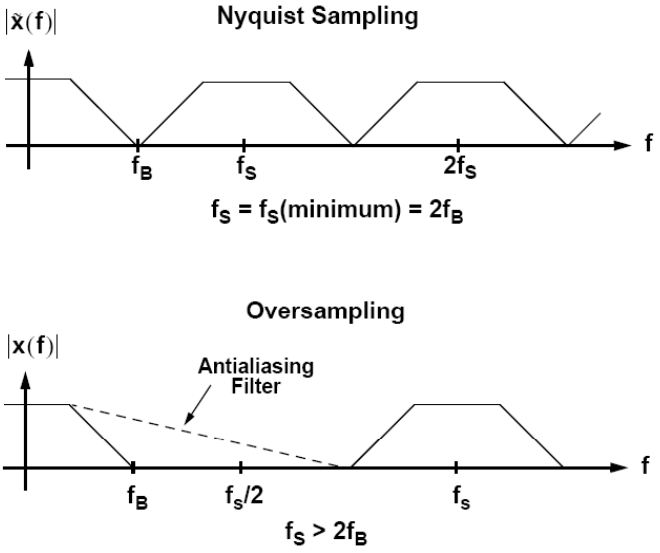


- Sampling theorem

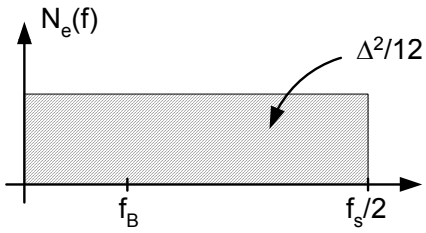
$$f_S > 2f_{sig,max}$$

- One good reason for sampling faster ("oversampling")
  - Can use lower order anti-alias filter

Anti-Alias Filtering

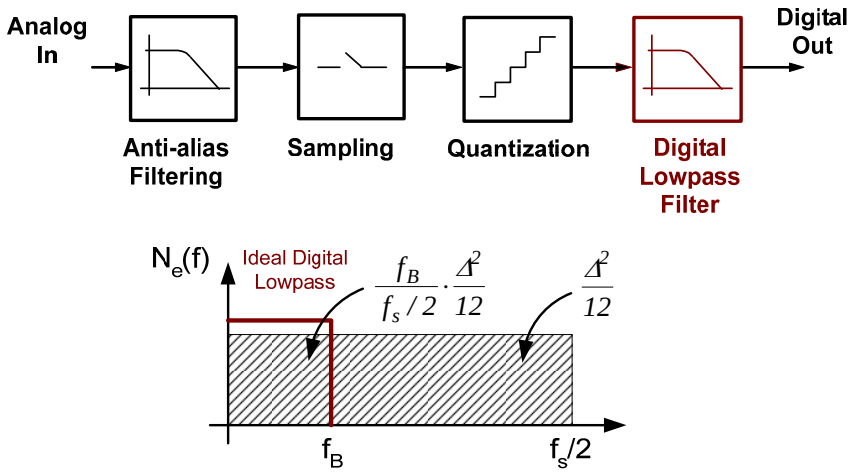


Quantization Noise



- Recall that the "noise" introduced by quantizer is evenly distributed across all frequencies
  - Provided that quantization error sequence is "sufficiently random"
- Idea: Let's filter out the noise beyond $f=f_B$!

Digital Noise Filter (1)



- Total quantization noise at digital output is reduced proportional to "oversampling ratio" $M=(f_s/2)/f_B$

Digital Noise Filter (2)

- Increasing M by 2x, means 3-dB reduction in quantization noise power, and thus 1/2 bit increase in resolution
  - "1/2 bit per octave"
- Is this useful?
- Reality check
  - Want 16-bit ADC, f_B=1MHz
  - Use oversampled 8-bit ADC with digital lowpass filter
  - 8-bit increase in resolution necessitates oversampling by 16 octaves

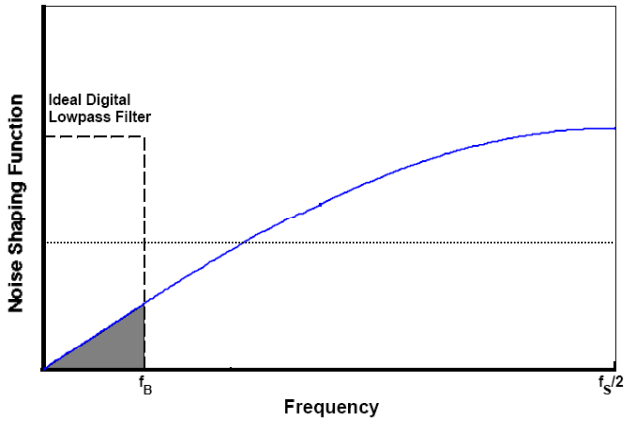
$$f_s \geq 2 \cdot f_B \cdot M = 2 \cdot 1MHz \cdot 2^{16}$$

$$\geq 131GHz$$

Noise Shaping Modulators

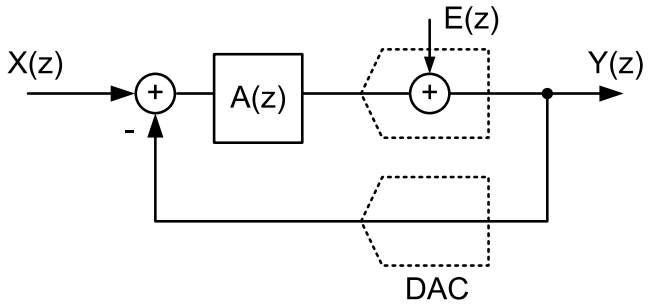
- Sample and coarsely quantize the input at a rate well above the Nyquist rate
- Shape the spectrum of the quantization noise so as to push most of its energy outside the signal baseband
- Out-of-band noise, including quantization noise, is suppressed by a subsequent digital lowpass filter (ie decimation filter)
- Output of the digital filter can be resampled at a lower sampling rate if the filter provides adequate anti-aliasing, as well as noise suppression

Noise Shaping



- Idea: "Somehow" build an ADC that has most of its quantization noise at high frequencies
- Key: Feedback

Noise Shaping Using Feedback (1)



$$Y(z) = E(z) + A(z)X(z) - A(z)Y(z)$$

$$= E(z)\frac{1}{1+A(z)} + X(z)\frac{A(z)}{1+A(z)}$$

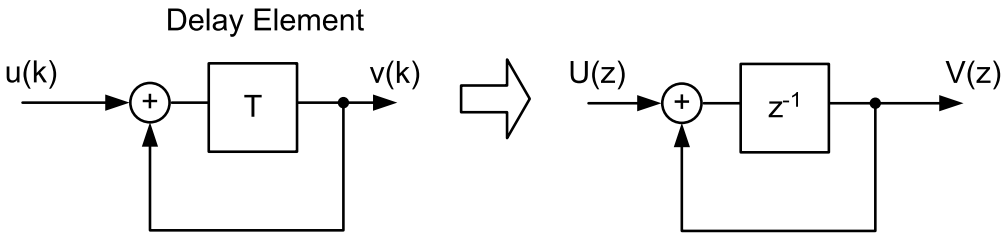
$$= E(z)\underbrace{H_E(z)}_{\substack{\text{Noise}\\ \text{Transfer}\\ \text{Function}}} + X(z)\underbrace{H_X(z)}_{\substack{\text{Signal}\\ \text{Transfer}\\ \text{Function}}}$$

Noise Shaping Using Feedback (2)

$$Y(z) = E(z)\underbrace{\frac{1}{1+A(z)}}_{\substack{\text{Noise}\\ \text{Transfer}\\ \text{Function}}} + X(z)\underbrace{\frac{A(z)}{1+A(z)}}_{\substack{\text{Signal}\\ \text{Transfer}\\ \text{Function}}}$$

- Objective
  - Want to make STF unity in the signal frequency band
  - Want to make NTF "small" in the signal frequency band
- If the frequency band of interest is around DC (0…f_B) we achieve this by making |A(z)| >>1 at low frequencies
  - Means that NTF is <<1
  - Mans that STF ≅ 1

Discrete Time Integrator

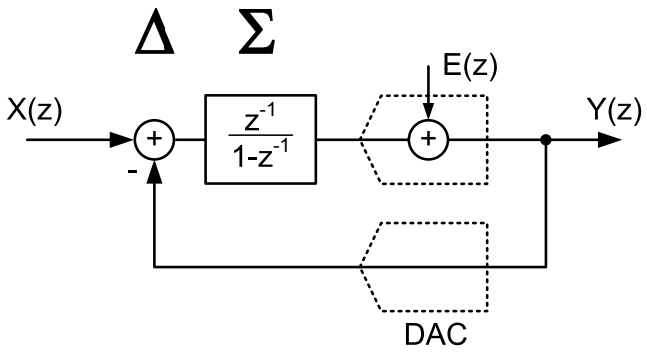


$$v(k) = u(k-1) + v(k-1) \qquad V(z) = z^{-1}U(z) + z^{-1}V(z)$$

$$\frac{V(z)}{U(z)} = \frac{z^{-1}}{1-z^{-1}} = \frac{1}{z-1} \qquad z = e^{j\omega T}$$

- "Infinite gain" at DC (ω=0, z=1)

First Order "Delta-Sigma" Modulator



$$Y(z) = E(z)\frac{1}{1+\frac{1}{z-1}} + X(z)\frac{\frac{1}{z-1}}{1+\frac{1}{z-1}} = E(z)\left(1-z^{-1}\right) + X(z)z^{-1}$$

- Output is equal to delayed input plus filtered quantization noise

NTF Frequency Domain Analysis

$$H_e(z) = 1 - z^{-1}$$

$$H_e(j\omega) = \left(1 - e^{-j\omega T}\right) = 2e^{-j\omega T/2}\left(\frac{e^{j\omega T/2} - e^{-j\omega T/2}}{2}\right)$$

$$= 2e^{-j\frac{\omega T}{2}}\left(j\sin\left(\frac{\omega T}{2}\right)\right) = 2\sin\left(\frac{\omega T}{2}\right)e^{-j\frac{\omega T - \pi}{2}}$$

$$\left|H_e(f)\right| = 2\left|\sin\left(\pi f T\right)\right| = 2\left|\sin\left(\pi\frac{f}{f_s}\right)\right|$$

- "First order noise Shaping"
- Quantization noise is attenuated at low frequencies, amplified at high frequencies

In-Band Quantization Noise (1)

- Question: If we had an ideal digital lowpass, what would be the achieved SQNR as a function of oversampling ratio?
- Can integrate shaped quantization noise spectrum up to f_B and compare to full-scale signal

$$P_{qnoise} = \int_0^{f_B} \frac{\Delta^2}{12}\cdot\frac{2}{f_s}\cdot\left[2\sin\left(\pi\frac{f}{f_s}\right)\right]^2 df$$

$$\cong \int_0^{f_B} \frac{\Delta^2}{12}\cdot\frac{2}{f_s}\cdot\left[2\pi\frac{f}{f_s}\right]^2 df$$

$$\cong \frac{\Delta^2}{12}\cdot\frac{\pi^2}{3}\left[\frac{2f_B}{f_s}\right]^3 = \frac{\Delta^2}{12}\cdot\frac{\pi^2}{3}\frac{1}{M^3}$$

In-Band Quantization Noise (2)

- Assuming a full-scale sinusoidal signal, we have

$$SQNR \cong \frac{P_{sig}}{P_{qnoise}} = \frac{\frac{1}{2}\left(\frac{\left(2^B-1\right)\Delta}{2}\right)^2}{\frac{\Delta^2}{12}\cdot\frac{\pi^2}{3}\frac{1}{M^3}} = 1.5\times\left(2^B-1\right)^2\times\underbrace{\frac{3}{\pi^2}\times M^3}_{\text{Due to noise shaping \& digital filter}}$$

$$\cong 1.76+6.02B-5.2+30\log(M) \quad [dB] \quad \text{(for large B)}$$

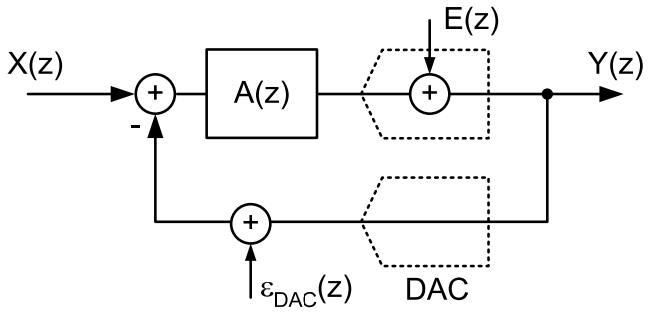
- Each 2x increase in M results in 8x SQNR improvement
  - 9dB (1.5bits) per octave oversampling

SQNR Improvement

- Example revisited
  - Want 16-bit ADC, f_B=1MHz
  - Use oversampled 8-bit ADC, first order noise shaping and (ideal) digital lowpass filter
    - SQNR improvement compared to case without oversampling is -5.2dB+30log(M)
  - 8-bit increase in resolution (48dB SQNR improvement) would necessitate M≅60
- Not all that bad!

M	SQNR improvement
16	31dB (~5 bits)
256	67dB (~11 bits)
1024	85dB (~14 bits)

DAC Requirements



$$Y(z) = E(z)\frac{1}{1+A(z)} + \left[X(z) - \varepsilon_{DAC}(z)\right]\frac{A(z)}{1+A(z)}$$

- DAC error is indistinguishable from signal
  - Means that DAC must be precise to within target resolution
- For the previous example, this means that we need an 8-bit DAC whose output levels have 16-bit precision…

Solutions

- Trimming or calibration
  - Measure DAC levels during test or at power-up
  - Apply correction values to each level using auxiliary DAC
- Dynamic Element Matching Algorithms
  - Shuffle DAC unit elements to obtain fairly precise "average" output levels
  - Two ways
    - Data independent shuffling
    - Data dependent shuffling
  - Data dependent shuffling algorithms allow to push most of the DAC "noise" outside the signal band
  - See e.g. [Carley, JSSC 4/1989], [Galton, TCAS II 10/1997], [Vleugels, JSSC 12/2001]
- Single bit DAC

Data Independent Shuffling (1)

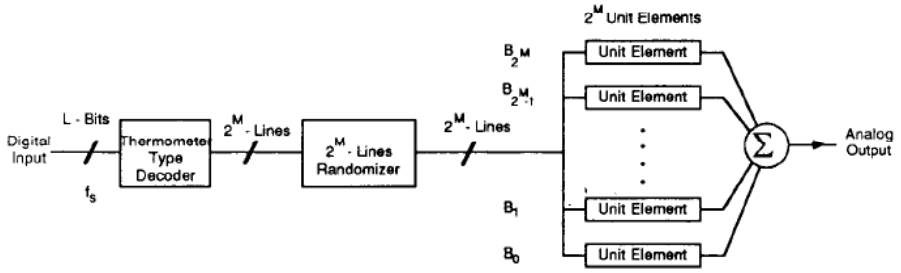
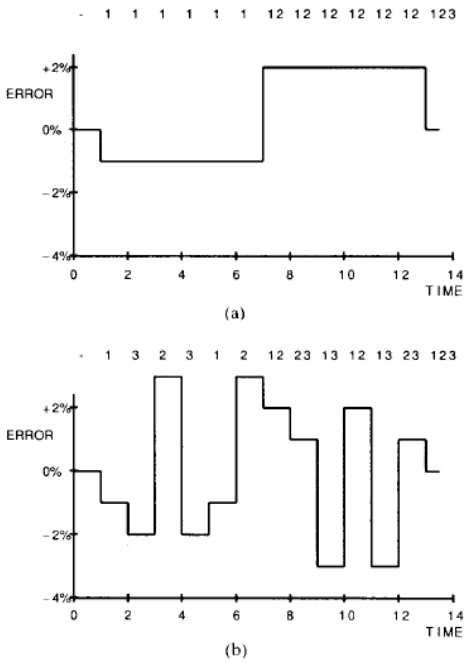


Fig. 3. Topology of dynamic element matching internal D/A converter. A thermometer-type decoder sets the number of output lines high that is equal to the digital input.

Carley, L.R., "A noise-shaping coder topology for 15+ bit converters," *IEEE JSSC*, vol.24, no.2, pp.267-273, Apr. 1989.

Data Independent Shuffling (2)

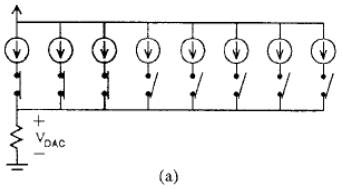


Carley, L.R., "A noise-shaping coder topology for 15+ bit converters," *IEEE JSSC*, vol.24, no.2, pp.267-273, Apr. 1989.

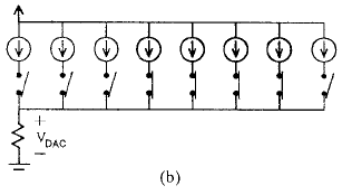
Data Dependent Shuffling (1)

- Select elements such that each one is used (on average) the same number of times
- DAC errors quickly sum to zero; errors are pushed to high frequencies

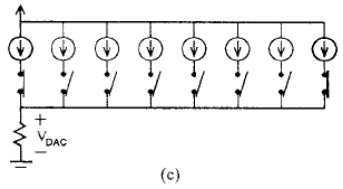
R.T. Baird, and T.S. Fiez, "Improved ΔΣ DAC linearity using data weighted averaging ," *IEEE ISCAS*, pp.13-16, May 1995.



(a) Code 3



(b) Code 4



(c) Code 2

Data Dependent Shuffling (2)

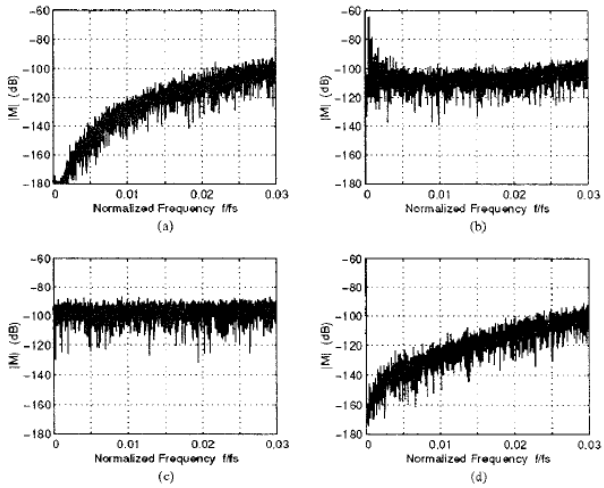
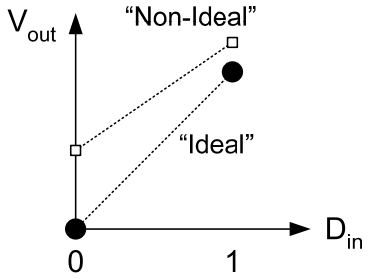


Figure 2: Simulated magnitude spectra for a third-order ΔΣ modulator with a 3-bit quantizer using an (a) ideal DAC, (b) DAC with 1% component mismatches, (c) randomization DEM, (d) data weighted averaging DEM.

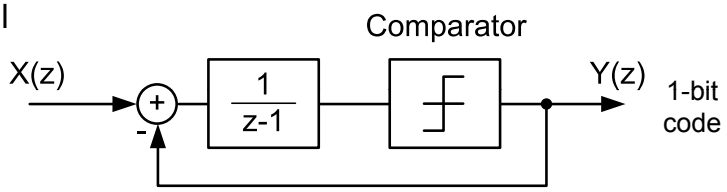
Single-Bit DAC

- A single bit DAC has only two output levels
- Even if these two levels are imprecise, the errors will only affect gain and offset of the DAC and modulator
  - Tolerable in many applications



Modulator with Single-Bit Quantizer (1)

- Model



- Expected SQNR (from slide 15 with B=1)

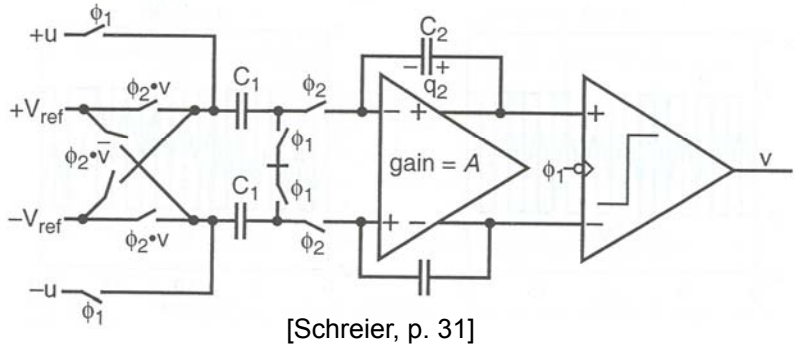
$$SQNR \cong \frac{P_{sig}}{P_{qnoise}} = \frac{\frac{1}{2}\left(\frac{\Delta}{2}\right)^2}{\frac{\Delta^2}{12}\cdot\frac{\pi^2}{3}\frac{1}{M^3}} = \frac{9}{2\pi^2}\times M^3$$

$$= -3.4 + 30\,log(M) \quad [dB]$$

- E.g. M=128 $\Rightarrow$ SQNR=60dB

Modulator with Single-Bit Quantizer (2)

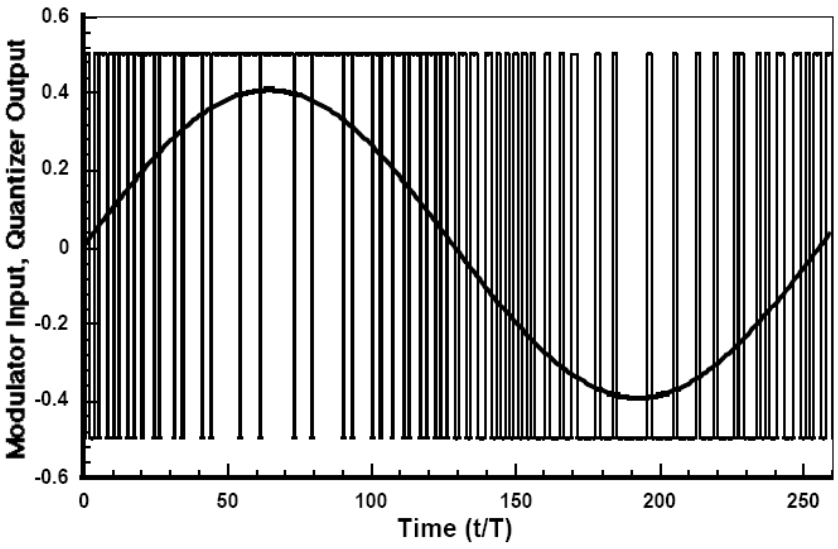
- Implementation example



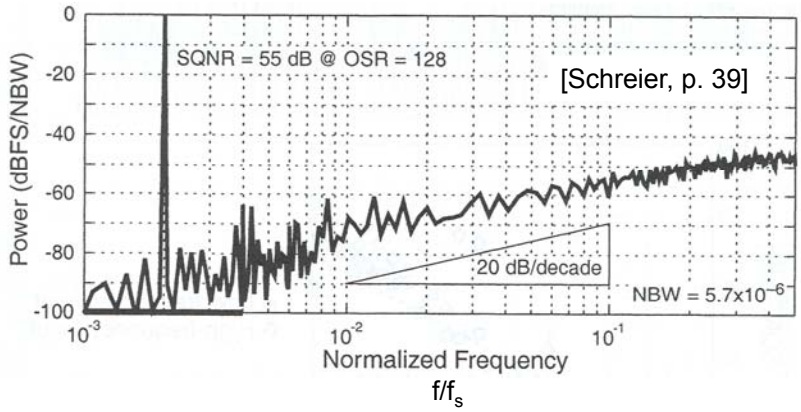
[Schreier, p. 31]

- Not all that great in terms of achievable SQNR, but sufficient for some applications
  - E.g. digital voltmeter
    - See [van de Plassche, pp. 469]

Simulated Response

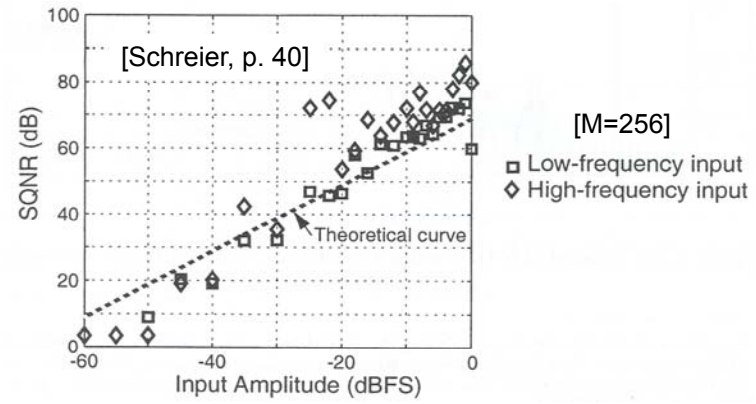


Spectrum



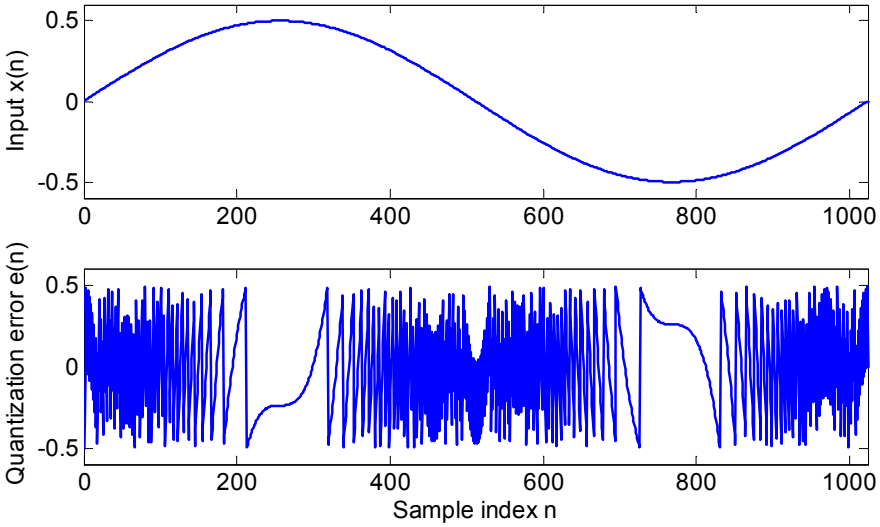
- Looks like there is some noise shaping, but SQNR=55dB is lower than the expected 60dB

Amplitude and Frequency Dependence



- Erratic dependence on amplitude and frequency
  - Simple linear model fails to predict this behavior
- Issue: Quantization error sequence is not "sufficiently random", as assumed in the beginning of this discussion

Quantization Error in 1st Order Modulator



- A complicated, but deterministic function of the input

Aside: Quantizer Gain

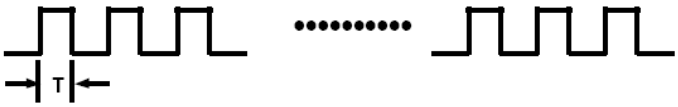
- Another issue is that the gain of the single bit quantizer is ill-defined, but we assumed it to be unity in our analysis
- The actual quantizer gain can be found from simulations, and then plugged back into the linear model for better agreement (and stability analysis using root locus, etc.)
- References
  - Schreier, Sections 3.2 and 4.2
  - S. Ardalan, and J. Paulos, "An analysis of nonlinear behavior in delta - sigma modulators," *IEEE TCAS*, vol.34, no.6, pp. 593-603, June 1987.
  - T. Ritoniemi, T. Karema, and H. Tenhunen, "Design of stable high order 1-bit sigma-delta modulators," *Proc. IEEE ISCAS,* pp. 3267-3270, May 1990.

Tones

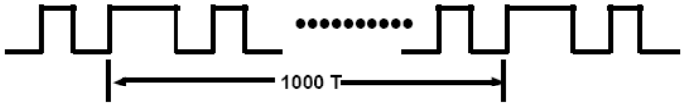
- Since the quantization error is correlated with the input, the shaped quantization noise contains spurious tones, some of which lie in the signal band
- The linear model cannot predict these tones
- It is generally difficult to predict tonal behavior for arbitrary inputs, even with a nonlinear model
  - Analytical results exist for DC and sine inputs, see e.g.
    - R.M. Gray "Spectral analysis of quantization noise in a single-loop sigma-delta modulator with DC input," IEEE Trans. Comm., pp. 588-599, June 1989.
    - R.M. Gray et al., Quantization noise in single-loop sigma-delta modulation with sinusoidal inputs," IEEE Trans. Comm., pp. 956-968, Sept 1989.
- Interesting and intuitive to look at DC input as a worst case

DC Input (1)

- E.g. x(n)=0
  - Modulator generates an alternating sequence of 1s and 0s
  - Single tone at $f_s/2$; no low frequency component



- E.g. x(n)=0.001·Δ/2
  - Compared to previous example, only one in 1000 outputs will change
  - Output has period of 1000·T, and hence contains a low frequency, in-band component



DC Input (2)

- For a DC input, the modulator output consists of discrete tones ("idle tones") with power and frequency given by

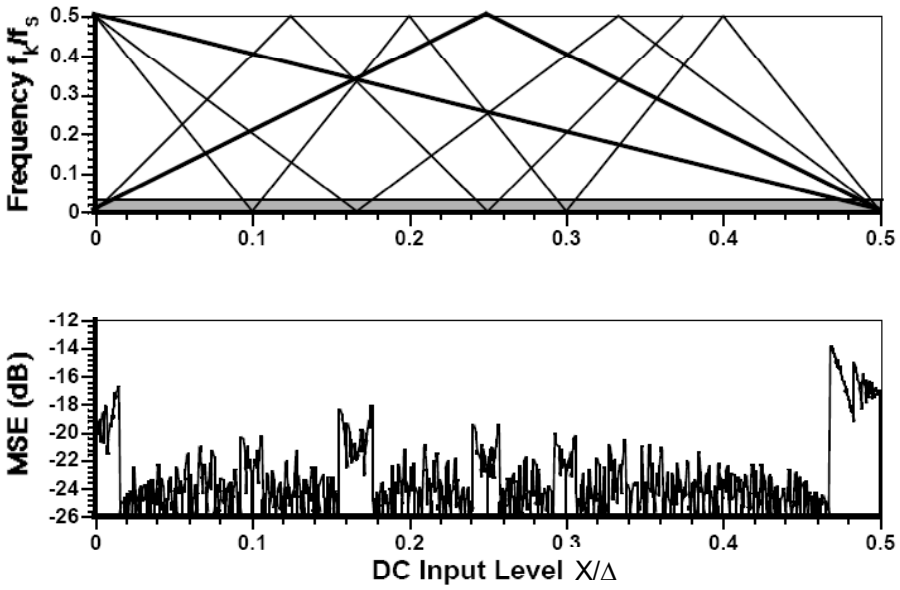
$$P_k = \left(\frac{\Delta \sin(\pi f_k T)}{\pi k} \right)^2$$

$$f_k = \left\langle k \left(\frac{x_{DC}}{\Delta} + 0.5 \right) \right\rangle f_s$$

  where k is an integer, and <r> represents the fractional part of r (r modulo 1)

- Strongest tones occur for small k, due to reciprocal dependence
- The plot on the following slide shows the total mean square error due to in-band idle tones as a function of DC input (M=16)

MSE due to Idle Tones



Idle Tone Considerations

- Idle tones are known to be a significant issue in audio applications
  - The human ear can detect tones ~20dB below the thermal/quantization noise floor
- If idle tones are an issue, there are several options for mitigating their impact
  - Larger oversampling ratio
  - Multi-bit quantizer and DAC
  - Dither
    - Superimpose a pseudorandom signal at the quantizer input to "whiten" quantization noise
      - See e.g. Chapter 3 of *Delta-Sigma Data Converters* by Norsworthy, Schreier & Temes.
  - Overdesign by making quantization noise much smaller than electronic noise from integrators
    - Noisy integrator(s) help randomize quantization error sequence
  - Higher order modulators
    - Naturally produce "more random" quantization error sequences

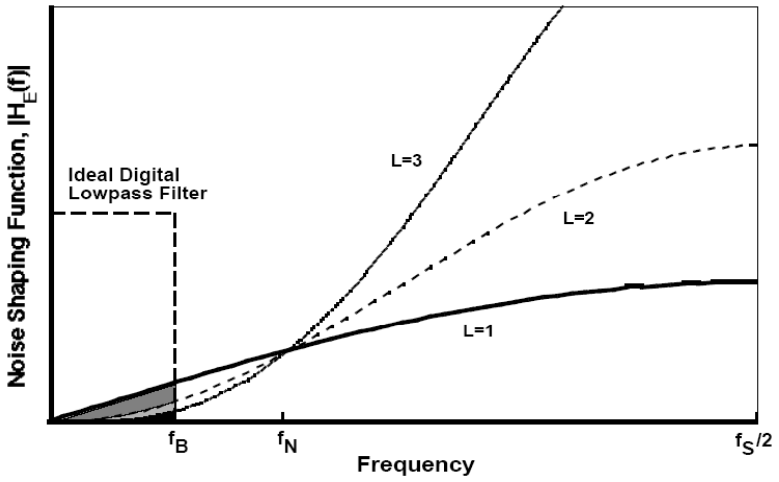
Higher Order Modulators

- Motivation: better SQNR for a given oversampling ratio, plus improved idle tone performance as a side benefit
- Commonly used architectures
  - Single quantizer loop with higher order filtering
    - Essentially a logical extension to the first order noise shaping concept discussed previously
  - Cascaded, multi-stage modulators
    - Contain a separate quantizer in each stage

Higher Order Noise Shaping

- L^{th} order noise transfer function

$$H_E(z) = \left(1 - z^{-1}\right)^L$$



In-Band Quantization Noise

$$P_{qnoise} = \int_0^{f_B} \frac{\Delta^2}{12} \cdot \frac{2}{f_s} \cdot \left[2 sin\left(\pi \frac{f}{f_s} \right) \right]^{2L} df$$

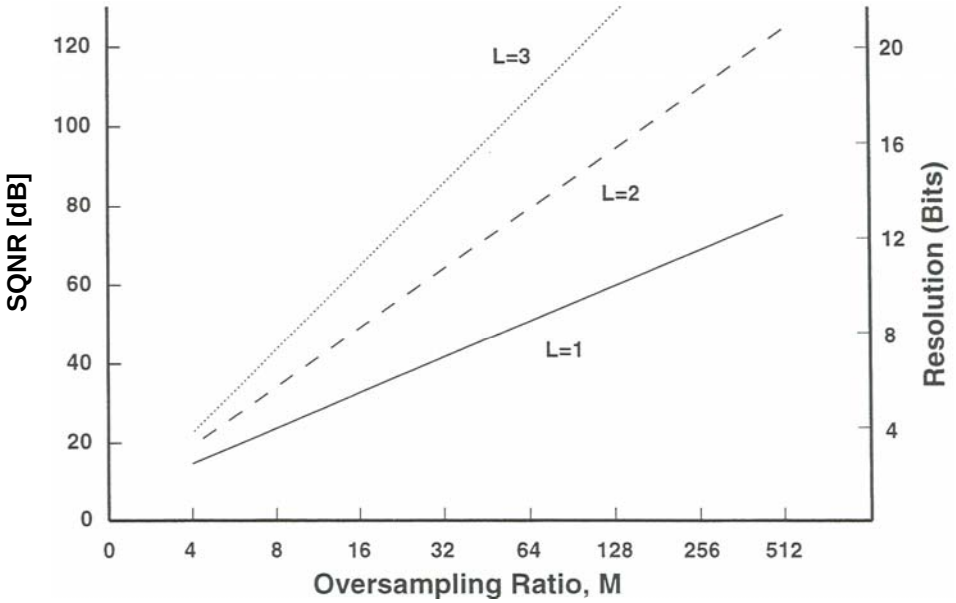
$$\cong \int_0^{f_B} \frac{\Delta^2}{12} \cdot \frac{2}{f_s} \cdot \left[2\pi \frac{f}{f_s} \right]^{2L} df$$

$$\cong \frac{\Delta^2}{12} \cdot \frac{\pi^{2L}}{2L+1} \left[\frac{2f_B}{f_s} \right]^{2L+1}$$

$$\cong \frac{\Delta^2}{12} \cdot \frac{\pi^{2L}}{2L+1} \left(\frac{1}{M} \right)^{2L+1}$$

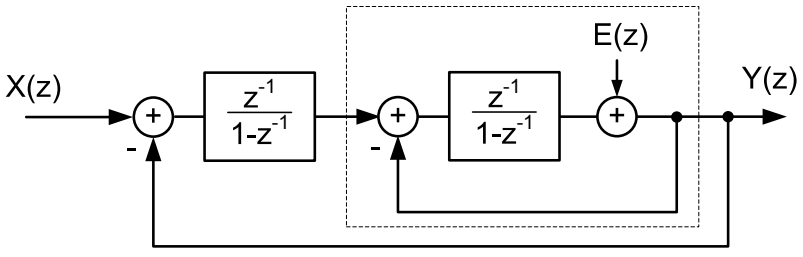
- For an L^{th} order modulator, every doubling of M results in an increase in SQNR of 6L+3dB (L+0.5bits)

SQNR with Single Bit Quantizer



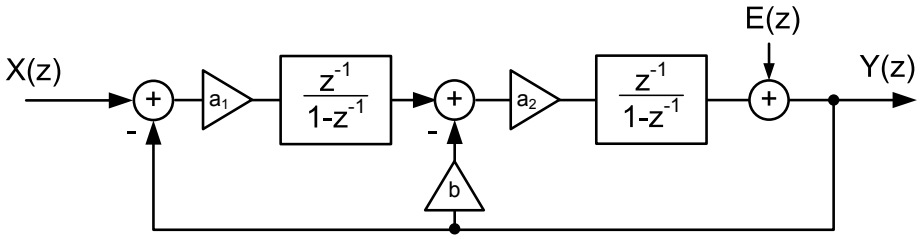
Building a Second-Order Modulator (1)

- General idea: Start with a first order modulator and replace quantizer by another first order loop



Building a Second-Order Modulator (2)

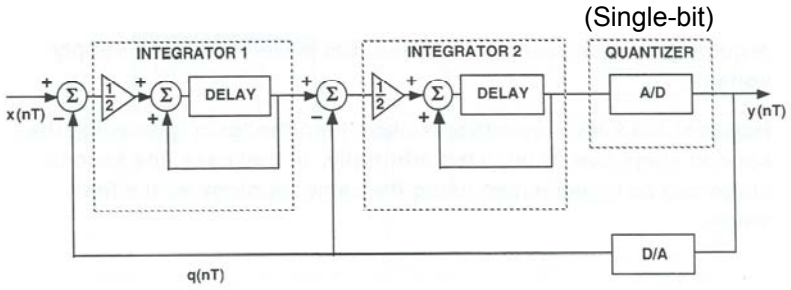
- More general structure



$$H_X(z) = \frac{a_1 a_2 z^{-2}}{D(z)} \qquad H_E(z) = \frac{\left(1-z^{-1}\right)^2}{D(z)}$$

$$D(z) = \left(1-z^{-1}\right)^2 + a_2 b z^{-1}\left(1-z^{-1}\right) + a_1 a_2 z^{-2} = 1 \quad \text{e.g. for} \quad \begin{cases} a_1 = a_2 = 1 \text{ and } b = 2 \\ a_1 = 0.5, a_2 = 2 \text{ and } b = 1 \end{cases}$$

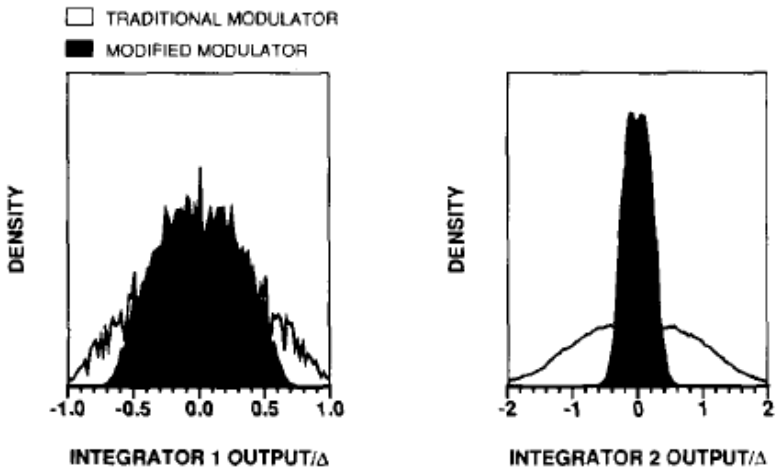
Boser-Wooley Modulator (1)



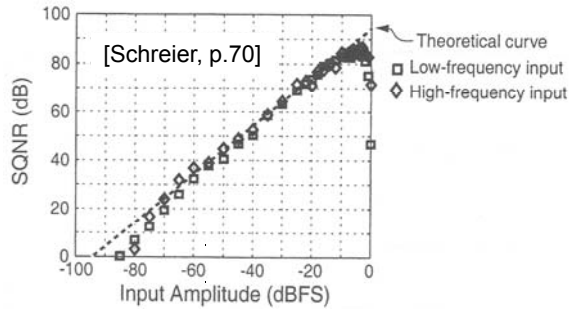
[Boser & Wooley, JSSC 12/1988]

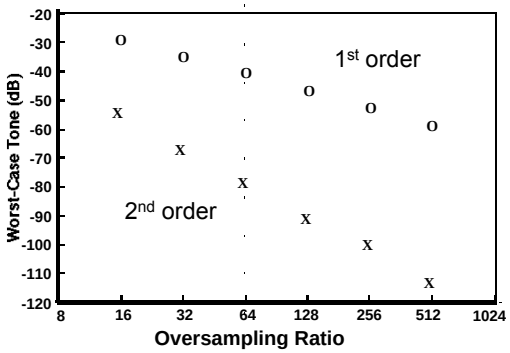
- a_1=0.5 and b=1, but a_2=0.5 (instead of 2)
- Since the integrator is followed by a single-bit quantizer, a_2 can be scaled to reduce swing requirements in second integrator

Boser-Wooley Modulator (2)



Performance of 2nd Order Modulator

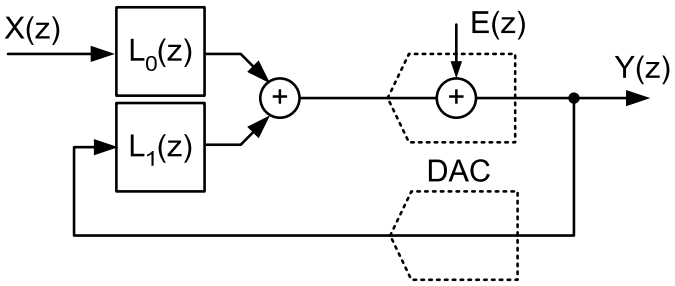




- Compared to first order modulator, SQNR is in "better" agreement with simple linear model
- Improved idle tone performance

Single Quantizer Modulators with Order >2

- Most general (mathematical) filter decomposition



$$H_e(z) = \frac{1}{1 - L_1(z)} \qquad H_x(z) = \frac{L_0(z)}{1 - L_1(z)}$$

$$L_0(z) = \frac{H_x(z)}{H_e(z)} \qquad L_1(z) = 1 - \frac{1}{H_e(z)}$$

Stability

- Having more than two integrators in a feedback loop means that the loop can be unstable (criterion = BIBO)
- From the diagram of the previous slide, it is clear that the stability of the loop mostly depends on $L_1(z)$, and therefore the characteristics of the NTF
- How about the nonlinear transfer characteristic of the quantizer?
  - Unfortunately, there is no crisp mathematical result that would address this question for all possible configurations
  - One important, and general aspect of having a nonlinearity in the loop is that the stability becomes dependent on the signal (and also L_0)!
- In practice, designers rely on a combination of stability analyses using the linear model (!), established “heuristics,” and time domain simulations of the nonlinear model

Stability Heuristics

- Single-bit
  - First order modulator is stable (bounded integrator output) with arbitrary inputs of less than $\Delta/2$ in magnitude
  - Second order modulator is known to be stable with arbitrary inputs of less than $\Delta/20$ in magnitude, and for "reasonable", slow varying inputs of magnitude $<0.8\cdot\Delta/2$, integrator outputs are "likely" to stay within bounds
  - Lee's criterion: modulator is "likely" to be stable if $\max[H_e(\omega)]<1.5$
- Multi-bit
  - A modulator with N^{th} order differentiation using an N+1 bit quantizer is stable for arbitrary inputs with amplitude less than half the quantizer input range (Schreier, p. 104)
  - …

Achievable SQNR

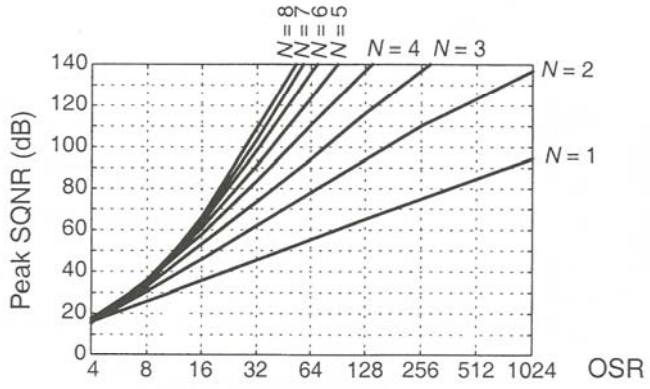
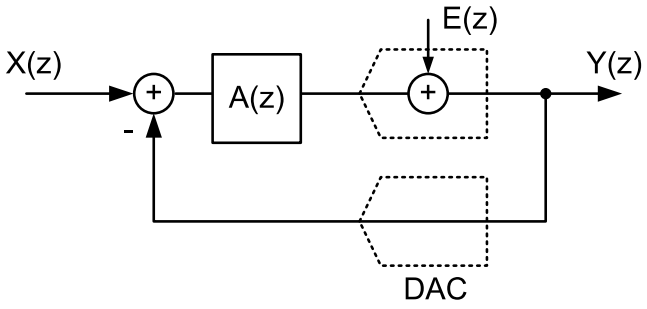


Figure 4.14: Empirical SQNR limit for 1-bit modulators of order N.

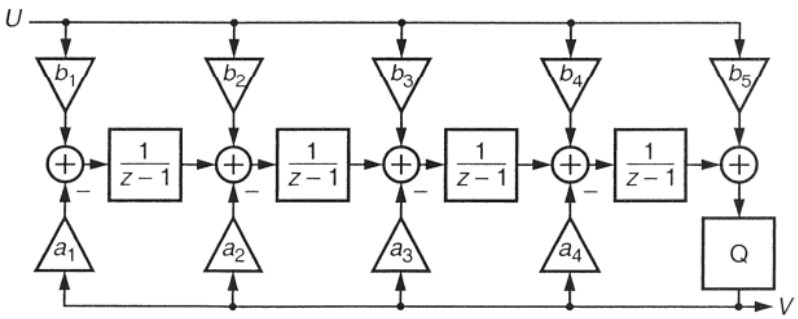
- Diminishing return for order greater 5-6

Topology Example: Single Feedback Loop



$$H_e(z) = \frac{1}{1 + A(z)} \qquad H_x(z) = \frac{A(z)}{1 + A(z)} \cong 1 \quad \text{in band of interest}$$

Topology Example: CIFB Architecture



[Schreier, p. 115]

- "Cascade of Integrators with Feedback"
- All zeros of NTF lie at DC for this structure, i.e. $H_E(z) = (1-z^{-1})^N$
- Can create complex conjugate zeros by adding feedback paths around pairs of integrators (cascade of resonators, CRFB structure)

Typical Design Procedure

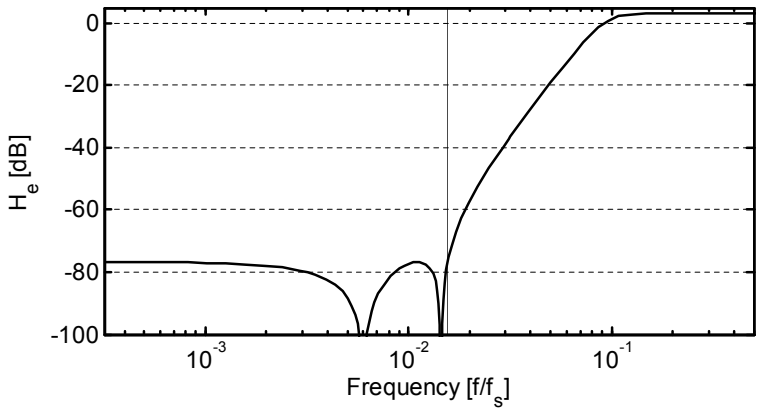
- "Cookbook design"
  - See e.g. *Delta-Sigma Data Converters*, by Norsworthy, Schreier & Temes, Sections 4.4 and 5.6
  - Choose order based on desired SQNR and M
  - Design NTF using filter approximations (e.g. Chebyshev2)
    - Make sure to obey Lee's criterion
  - Determine loop-filter transfer function and evaluate performance and stability using simulations
  - Determine implementation-specific coefficients
  - Scale coefficients to restrict integrator outputs to stay within available range ("dynamic range scaling")
- Delta-Sigma Toolbox for MATLAB (by Richard Schreier)
  - http://www.mathworks.com/matlabcentral/fileexchange
  - Look under "Controls" and find "Delsig" toolbox

"Cookbook" NTF Design Example (1)

```
% design parameters
L=4;  % order
M=64; % oversampling ratio

% stop-band attenuation; reduce if needed to make max(|He(w)|<1.5)
Rstop = 80;
[b,a] = cheby2(L, Rstop, 1/M, 'high');

% normalize to make He(z->inf)=1; needed for realizability
% makes first sample of impulse response of He equal to 1
% makes first sample of impulse response of A equal to 0
% (must have at least one delay around quantizer)
b = b/b(1);

% check Lee's rule; want max(|He(w)|<1.5 )
NTF = filt(b, a, 1)
[mag] = bode(NTF, pi)
```

"Cookbook" NTF Design Example (2)

```
Transfer function:
    1 - 3.998 z^-1 + 5.995 z^-2 - 3.998 z^-3 + z^-4
-------------------------------------------------------
1 - 3.247 z^-1 + 4.013 z^-2 - 2.231 z^-3 + 0.4699 z^-4

mag = 1.459
```



"Cookbook" NTF Design Example (3)

```
% Loop filter transfer function
A = inv(NTF) - filt(1,1,1)

Transfer function:
0.7505 z^-1 - 1.982 z^-2 + 1.766 z^-3 - 0.5301 z^-4
-------------------------------------------------------
  1 - 3.998 z^-1 + 5.995 z^-2 - 3.998 z^-3 + z^-4

% Check realizability
a = impulse(A);
a(1)

ans = 0
```

Commercial Example

ANALOG DEVICES

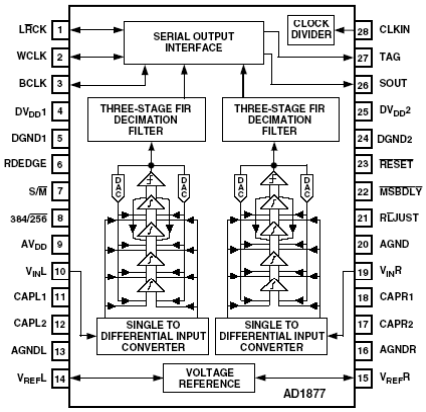
Single-Supply 16-Bit ΣΔ Stereo ADC

AD1877*

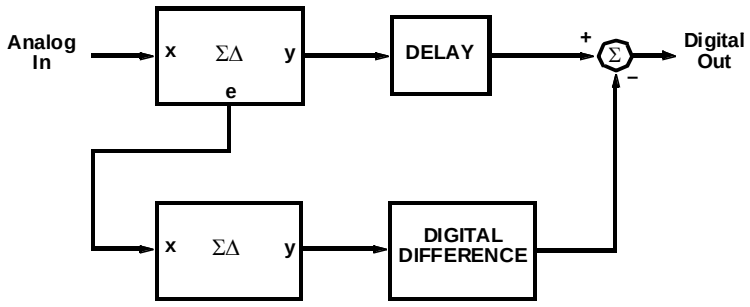
FEATURES
Single 5 V Power Supply
Single-Ended Dual-Channel Analog Inputs
92 dB (Typ) Dynamic Range
90 dB (Typ) S/(THD+N)
0.006 dB Decimator Passband Ripple
Fourth-Order, 64-Times Oversampling ΣΔ Modulator
Three-Stage, Linear-Phase Decimator
256 × F_S or 384 × F_S Input Clock
Less than 100 μW (Typ) Power-Down Mode
Input Overrange Indication
On-Chip Voltage Reference
Flexible Serial Output Interface
28-Lead SOIC Package

APPLICATIONS
Consumer Digital Audio Receivers
Digital Audio Recorders, Including Portables
CD-R, DCC, MD and DAT
Multimedia and Consumer Electronic Equipment
Sampling Music Synthesizers
Digital Karaoke Systems

FUNCTIONAL BLOCK DIAGRAM

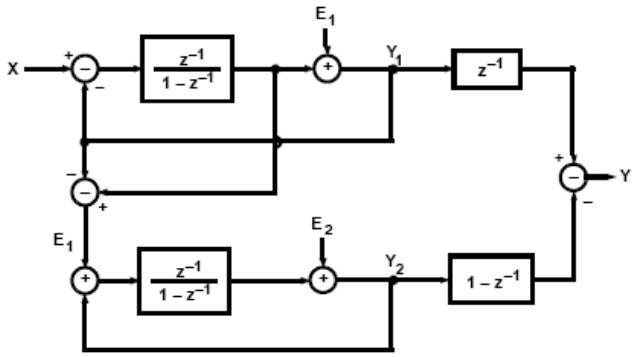


Cascaded Modulators



- Concept
  - Cascade of two or more stable (low order) modulators
  - Quantization error of each stage is quantized by the succeeding stages and subtracted in digital domain

Second Order (1-1) Cascade



$$Y_1(z) = z^{-1}X(z) + (1 - z^{-1})E_1(z)$$

$$Y_2(z) = z^{-1}E_1(z) + (1 - z^{-1})E_2(z)$$

$$Y(z) = z^{-1}Y_1(z) - (1 - z^{-1})Y_2(z)$$

$$= z^{-2}X(z) + z^{-1}(1 - z^{-1})E_1(z) - z^{-1}(1 - z^{-1})E_1(z) - (1 - z^{-1})^2E_2(z)$$

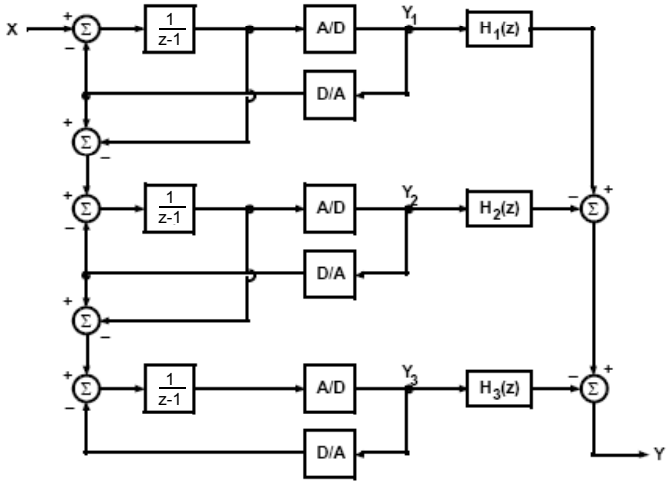
$$Y(z) = z^{-2}X(z) - (1 - z^{-1})^2E_2(z)$$

- Second order noise shaping using two first order loops!

Properties

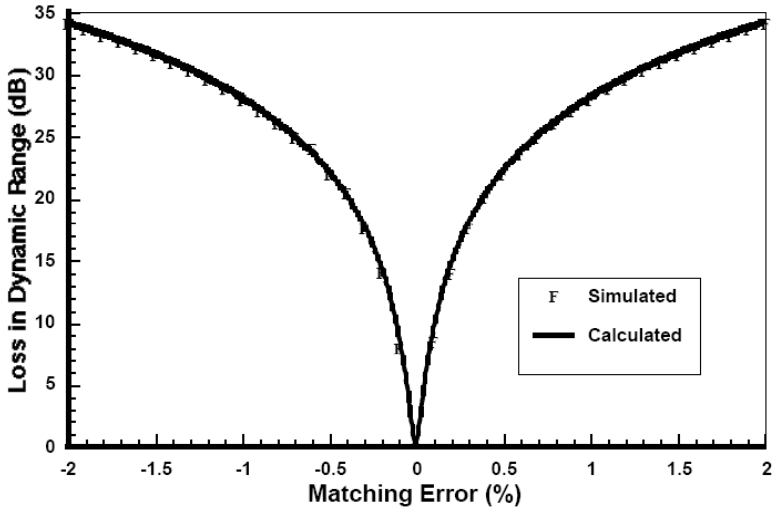
- Order of overall noise shaping is equal to sum of modulator orders
- No stability issues
- Improved idle tone performance
  - Input of second stage is "noise like"
  - Remaining quantization error from second stage is very close to white noise
- Cancellation of first stage quantization noise depends on matching between analog and digital signal paths
  - Hard to suppress first stage quantization error by more than 40dB
  - Mismatch will affect idle tone performance

1-1-1 Cascaded Modulator (MASH)

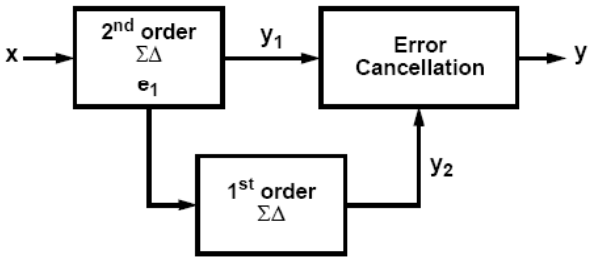


Mismatch Sensitivity

Sensitivity of 1-1-1 cascade to matching between the analog and digital "gains"



2-1 Cascade



$$Y_1(z) = z^{-2}X(z) + (1 - z^{-1})^2E_1(z)$$

$$Y_2(z) = z^{-1}E_1(z) + (1 - z^{-1})E_2(z)$$

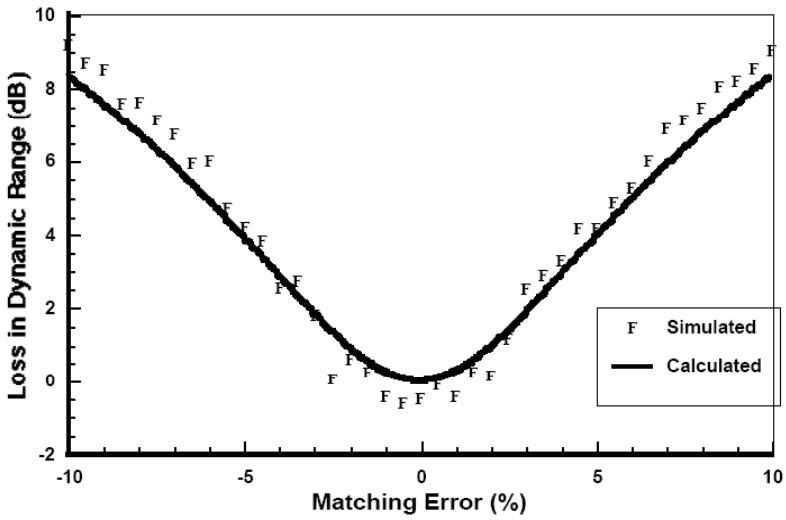
$$Y(z) = z^{-1}Y_1(z) - (1 - z^{-1})^2Y_2(z)$$

$$= z^{-3}X(z) + z^{-1}(1 - z^{-1})^2E_1(z) - z^{-1}(1 - z^{-1})^2E_1(z) - (1 - z^{-1})^3E_2(z)$$

$$Y(z) = z^{-3}X(z) - (1 - z^{-1})^3E_2(z)$$

Mismatch Sensitivity

Sensitivity of 2-1 cascade to matching error

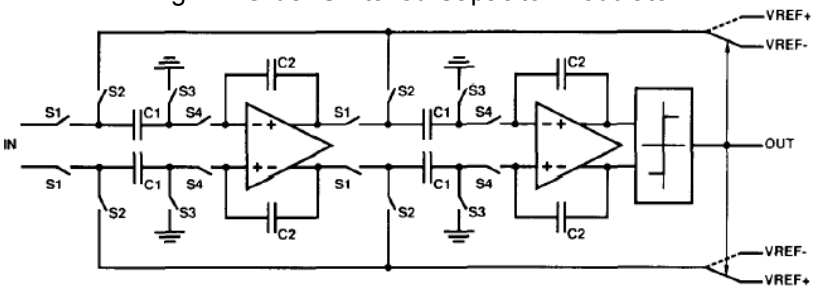


Circuit Level Considerations

- Electronic noise
- Integrator leak
  - Finite OTA gain
- Integrator nonlinearity
  - OTA dynamic settling error
  - OTA slewing
  - Capacitor voltage coefficients
- Comparator hysteresis
  - Usually not a problem; simulations show that up to a few % hysteresis can be tolerated
- Unwanted mixing effects
  - E.g. if V_{ref} contains $f_s/2$, out of band noise will be mixed down into signal band

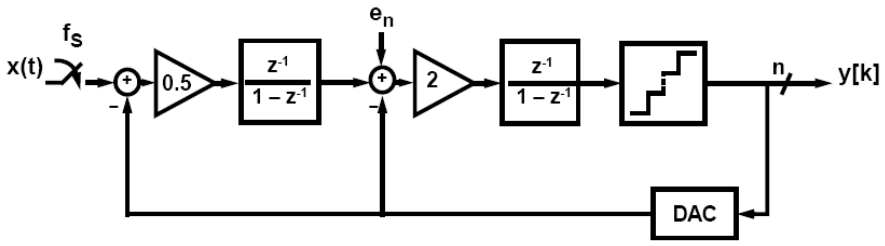
Electronic Noise

E.g. 2nd Order Switched Capacitor Modulator



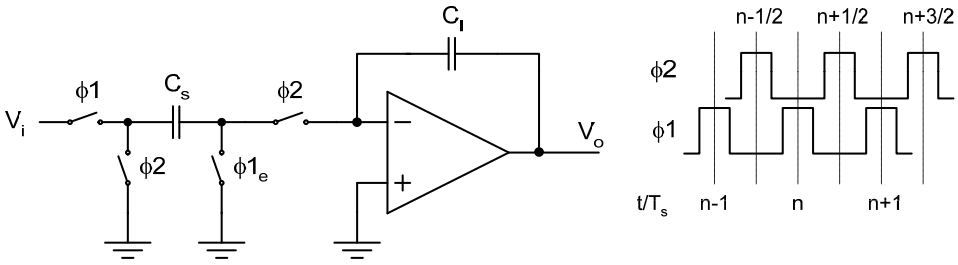
- Noise from 1st integrator is added directly to the input
  - Digital filter will reduce this noise by oversampling ratio
- Noise from 2nd integrator is first-order noise shaped!
  - Digital filter will remove most of this noise
- Especially for high oversampling ratios, only the first one or two integrators add significant noise
  - Qualitatively, this also holds for other imperfections.

Example – Noise from Second Integrator



Can show: $Y(z) = 2\left(1 - z^{-1}\right) z^{-1} e_n.$

Integrator Analysis (1)



t/T_s	Q_s	Q_I
n-1	$C_s \cdot V_i(n-1)$	$C_I \cdot V_o(n-1)$
n-1/2	0	$C_I \cdot V_o(n-1/2) = C_I \cdot V_o(n-1) + C_s \cdot V_i(n-1)$
n	$C_s \cdot V_i(n)$	$C_I \cdot V_o(n) = C_I \cdot V_o(n-1) + C_s \cdot V_i(n-1)$
n+1/2	…	…

Integrator Analysis (1)

- Assuming that V_o is sampled during $\phi 1$, we have

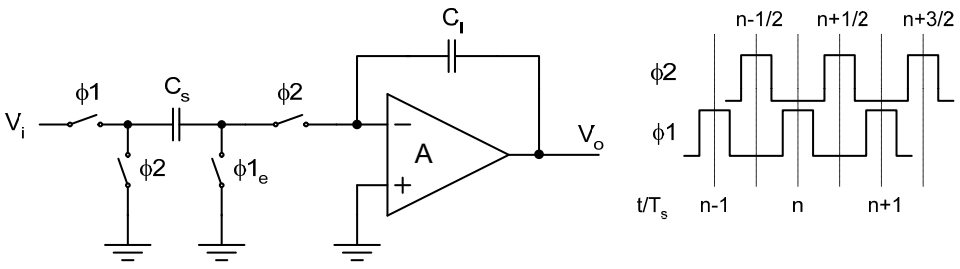
$$C_I V_o(n) = C_I V_o(n-1) + C_S V_i(n-1)$$

$$C_I V_o(z) = z^{-1} C_I V_o(z) + z^{-1} C_S V_i(z)$$

$$\therefore \frac{V_o(z)}{V_i(z)} = \frac{C_S}{C_I} \frac{z^{-1}}{1 - z^{-1}}$$

- Unfortunately, this ideal expression holds only for infinite amplifier gain
  - Let's look at impact of finite gain

Integrator Leak (1)



t/T_s	Q_s	Q_I
n-1	$C_s \cdot V_i(n-1)$	$C_I \cdot V_o(n-1) \cdot [1+1/A]$
n-1/2	$C_s \cdot V_o(n-1/2)/A$	$C_I \cdot V_o(n-1/2) \cdot [1+1/A] = C_I \cdot V_o(n-1) \cdot [1+1/A] + C_s \cdot V_i(n-1) - C_s \cdot V_o(n-1/2)/A$
n	$C_s \cdot V_i(n)$	$C_I \cdot V_o(n) \cdot [1+1/A] = C_I \cdot V_o(n-1) \cdot [1+1/A] + C_s \cdot V_i(n-1) - C_s \cdot V_o(n)/A$
n+1/2	…	…

Integrator Leak (2)

- Again, assuming that V_o is sampled during $\phi 1$, we have

$$C_I V_o(z)\left[1+\frac{1}{A}\right] = z^{-1} C_I V_o(z)\left[1+\frac{1}{A}\right] + z^{-1} C_S V_i(z) - \frac{C_S}{A} V_o(z)$$

$$\therefore \frac{V_o(z)}{V_i(z)} \cong \frac{C_S}{C_I} \frac{z^{-1}\left(1-\frac{1}{A}\left[1+\frac{C_S}{C_I}\right]\right)}{1-\left(1-\frac{1}{A}\frac{C_S}{C_I}\right)z^{-1}} = \frac{g \cdot z^{-1}}{1-[1-\alpha]\cdot z^{-1}}$$

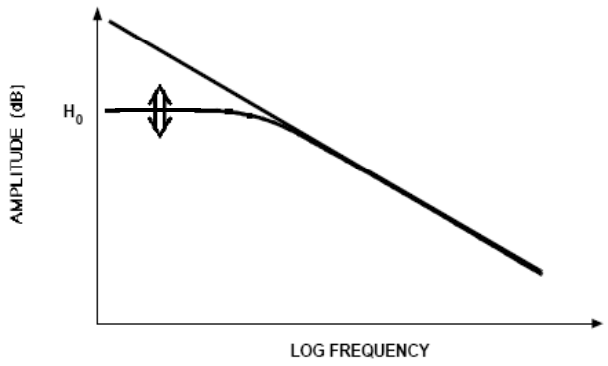
$$V_o(z) = [1-\alpha]\cdot z^{-1} V_o(z) + g \cdot z^{-1} V_i(z)$$

- Finite gain results in "leaky integrator"
  - Some fraction of previous output is lost in new cycle

Frequency Domain View

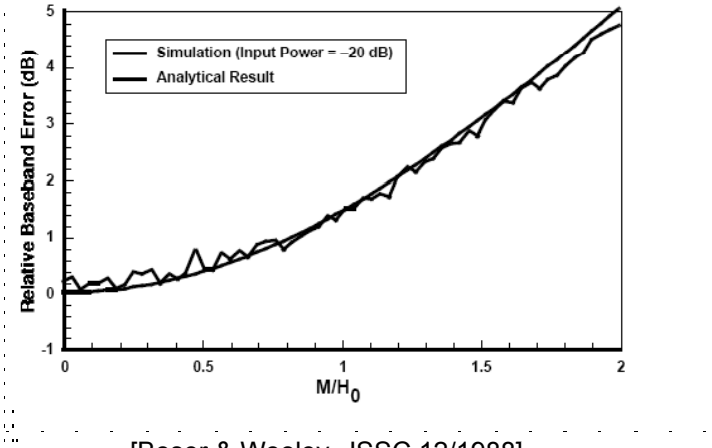
- Limited gain at low frequencies ($\omega \rightarrow 0$, $z \rightarrow 1$)

$$H_0 = H(z)\big|_{z=1} = \frac{g}{1-[1-\alpha]} = \frac{g}{\alpha} \propto A$$



- But noise shaping relies on high integrator gain at low frequencies…

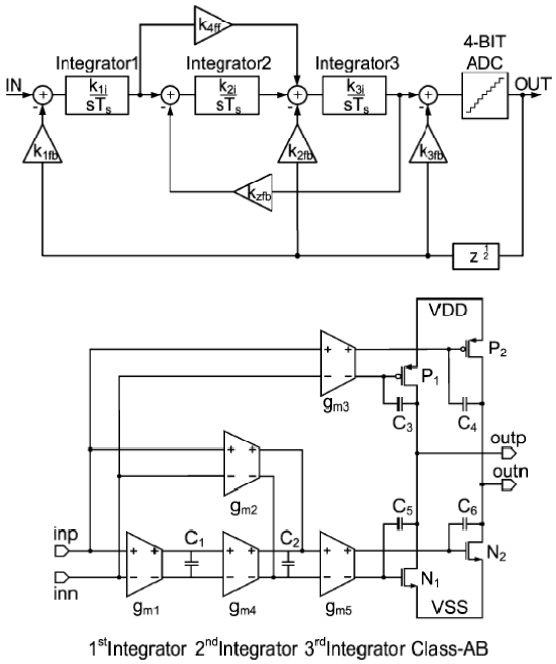
Required DC Gain



[Boser & Wooley, JSSC 12/1988]

- Good practice to make OTA gain at least a few times larger than oversampling ratio

State-of-the-art Continuous Time Modulator



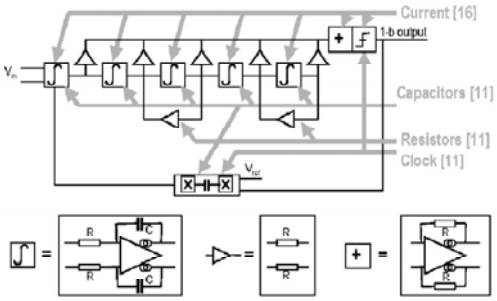
1st Integrator 2nd Integrator 3rd Integrator Class-AB

[Mitteregger, ISSCC 2006]

Sampling Frequency		640MHz
Conversion Rate		40MS/s
Input Range		0-20MHz
Peak SNR		76dB
THD		-78dB
Peak SNDR		74dB
ENOB		12
Process		1.2V 130nm 1P8M CMOS
Chip Area		8.6mm^2
Power	Modulator	20mW
	Decimator 40MS/s	18mW
	PLL 2.56GHz	12mW
	I/O 1.8V	4mW

- 4-stage amplifier with feedforward compensation
  - Impractical for SC circuits
  - Great for CT delta-sigma modulators

Multi-Mode Modulator



Process	1P 6M standard 90nm			
Supply voltage	1.1V – 1.3V (±3dB DR performance deviation)			
ΔΣ modulator	5th-order CT, feedforward, 1-b with SC DAC			
Input voltage range	$0.45V_{rms}$ differential			
Modes	121	GSM	BT	WLAN
Sampling rate	13MHz - 400MHz	26MHz	200MHz	400MHz
Signal bandwidth	100kHz - 10MHz	200kHz	1MHz	10MHz
Dynamic range	52dB - 82dB	82dB	75dB	52dB
Intermod. distances	IM2 > 70dB IM3 > 75dB			
Image Rejection	> 50dB			
Power@1.2V, one ΔΣ modulator	1.44mW - 7mW	1.44mW	3.4mW	7mW
FOM	0.2pJ/conv. - 0.8pJ/conv.	0.2pJ/conv.	0.31pJ/conv.	0.8pJ/conv.

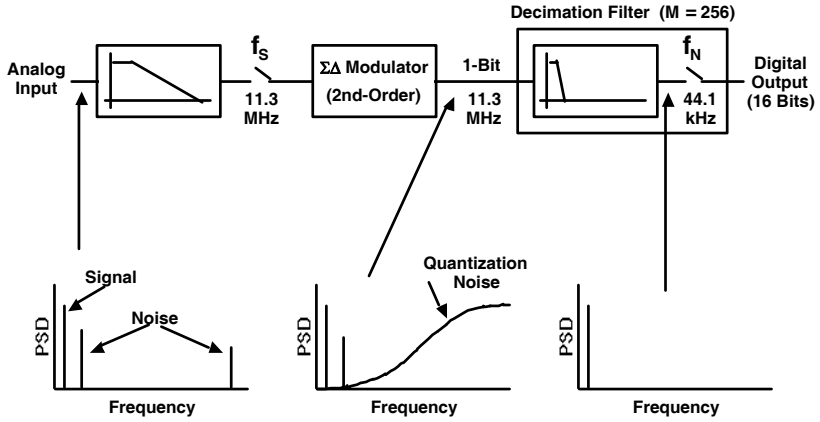
[Ouzounov, ISSCC 2007]

- Delta-sigma ADCs are more amenable to BW and DR reconfiguration
  - Very hard to reconfigure pipelined ADCs
- Great for flexible, multi-standard wireless receivers

Decimation Filters

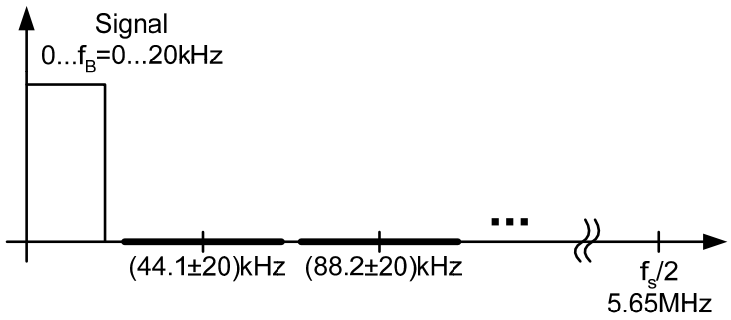
- References
  - J. Candy, "Decimation for Sigma-Delta Modulation," *IEEE Trans. Communications*, pp. 72-76, Jan. 1986.
  - Chapters 1 and 13 of *Delta-Sigma Data Converters*, by Norsworthy, Schreier, Temes.
  - B.P. Brandt and B.A. Wooley, "A low-power, area-efficient digital filter for decimation and interpolation," *IEEE J. Solid-State Circuits*, pp. 679-687, June 1994.
  - E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoustics, Speech and Signal Processing*, pp. 155-162, Apr 1981.
- Objectives
  - Remove out-of band quantization noise
  - Re-sample at lower frequency
    - Ideally at Nyquist rate

Example



- Filter must attenuate spectral components around ± N·f_N,
  - Otherwise they will alias onto signal after re-sampling

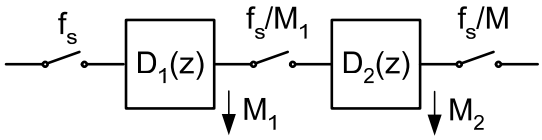
Filter Requirements



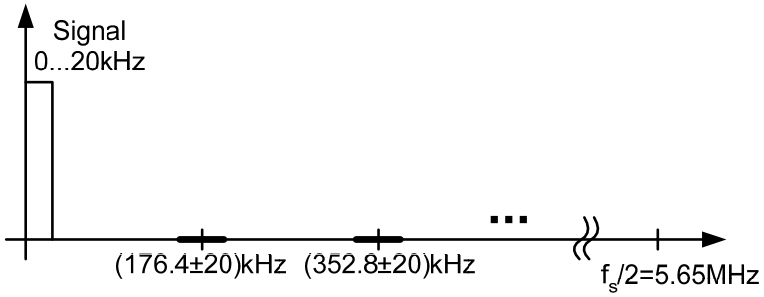
- Pass band 0…20kHz, transition band 20…24.1kHz (Δf=4.1kHz), stop band 24.1kHz…5.65MHz
- A digital FIR filter that meets these requirements would require more than $f_s/\Delta f$ = 11.3MHz/4.1kHz $\cong$ 2800 coefficients
  - Impractical!

Multi-Step Decimation

- Key idea: Don't try to decimate down to f_N in one step
  - Perform a gradual reduction of sampling rate + some filtering
  - E.g. Two-step decimation



- Example: M_1=64, f_s/M_1=176.4kHz



Sinc Filter (1)

- A popular, low complexity choice for stage 1 is the so-called sinc-filter
- From a time domain perspective, this filter simply computes the average of several samples

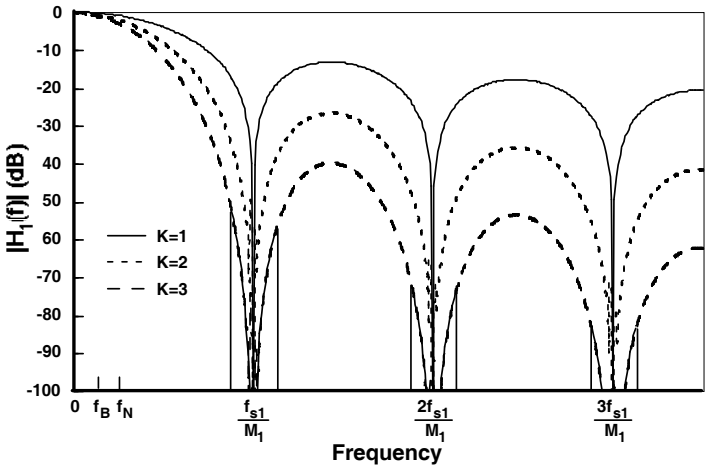
$$y(n) = \frac{1}{N}\sum_{i=0}^{N-1} x(n-i)$$

- Frequency domain

$$H(z) = \frac{1}{N}\frac{1-z^{-N}}{1-z^{-1}} \qquad H(\omega) = \frac{1}{N}\frac{\sin\left(\pi N \frac{f}{f_s}\right)}{\pi\frac{f}{f_s}} e^{-j\pi\frac{f}{f_s}(N-1)}$$

- Zeros at multiples of f_s/N
  - Make N=M_1 to attenuate alias components!

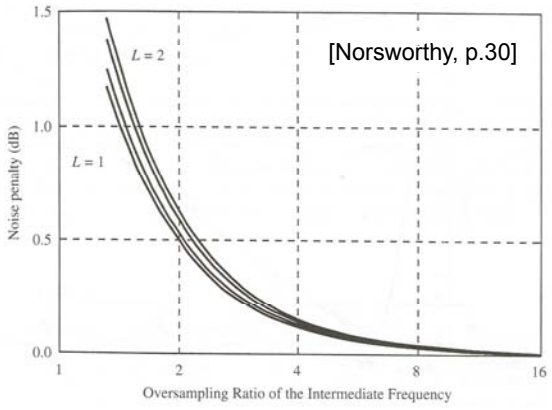
Cascade of K Sinc Filters



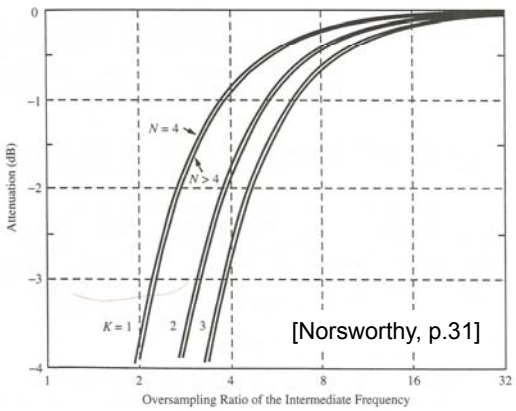
- Higher order means better rejection
  - But also more in-band droop
- Can show that for L^{th} order noise shaping, an $(L+1)^{th}$ order sinc filter is the best choice

Sinc Filter Performance

Noise penalty relative to "brick wall" filter



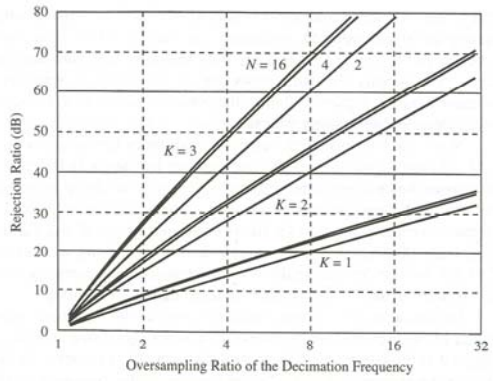
Droop



- Only about 0.14 dB increase in baseband noise for decimation to an intermediate oversampling ratio of 4
- If droop is undesired, it can be corrected downstream, using a separate post-emphasis filter

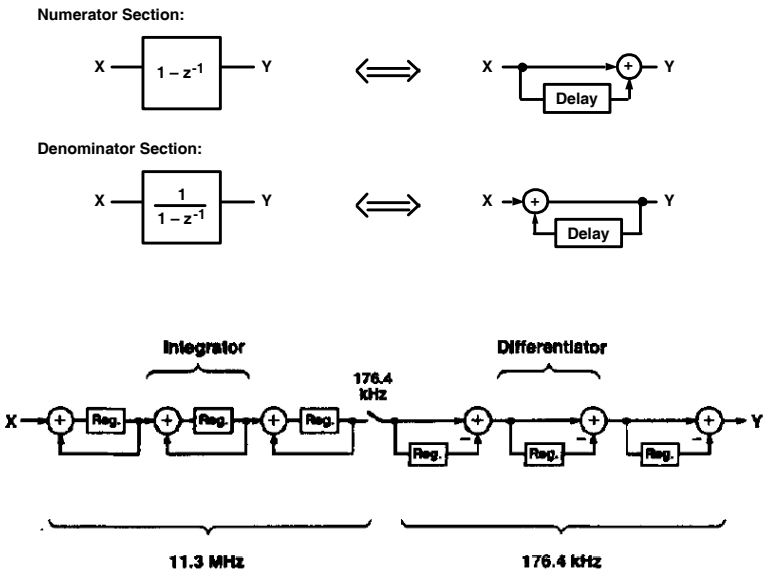
Sinc Filter Performance (2)

- In addition to suppressing quantization noise, the filter must attenuate out-of-band signals present at the modulator input
  - Worst case freqeuncy is f_s/M_1-f_B
  - E.g. 50dB for $sinc^3$, and intermediate oversampling of 4x
  - Any additional desired rejection must come from analog filter at modulator input

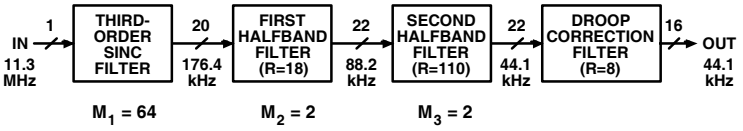


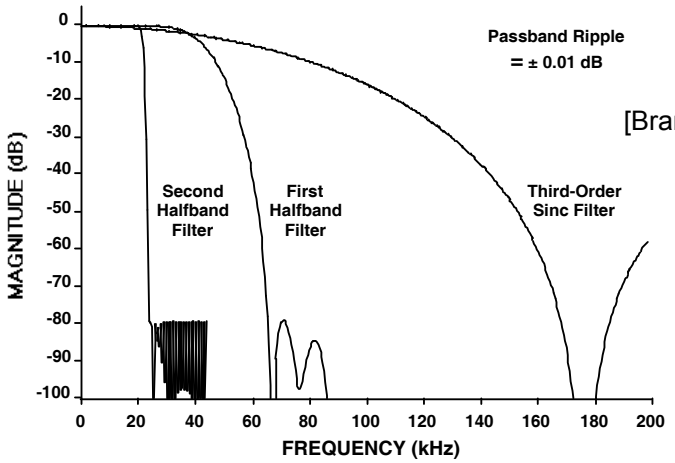
[Norsworthy, p.31]

Sinc Filter Implementation



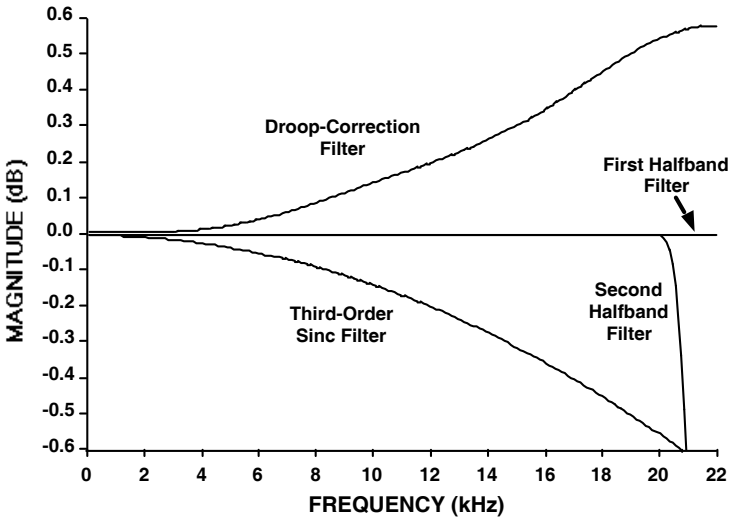
Complete Filter Implementation



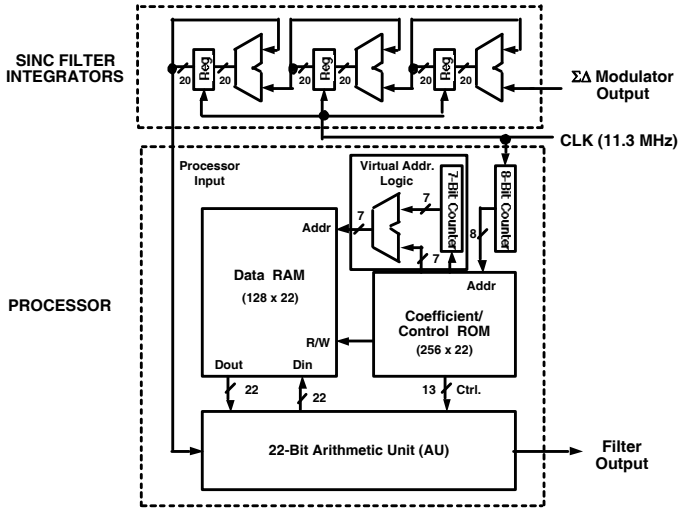


[Brandt & Wooley, JSSC 6/1994]

Droop Correction

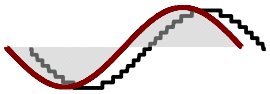


Implementation



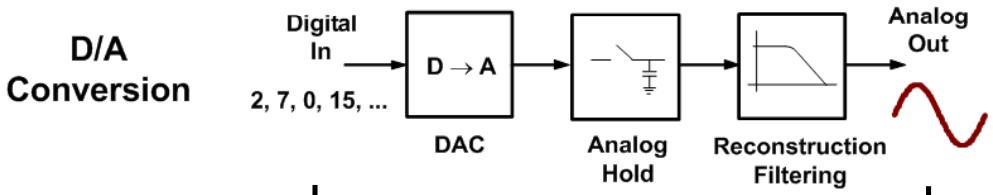
- 43 multiplications and 84 additions per output sample
- Can use serial arithmetic to minimize hardware area
  - Since output rate is usually fairly low

Oversampling D/A Conversion



Katelijn Vleugels
Stanford University

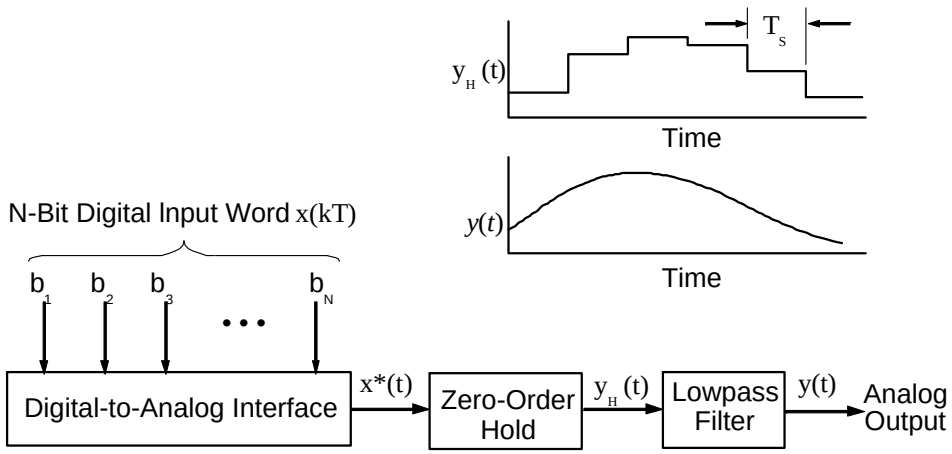
Recap (1)



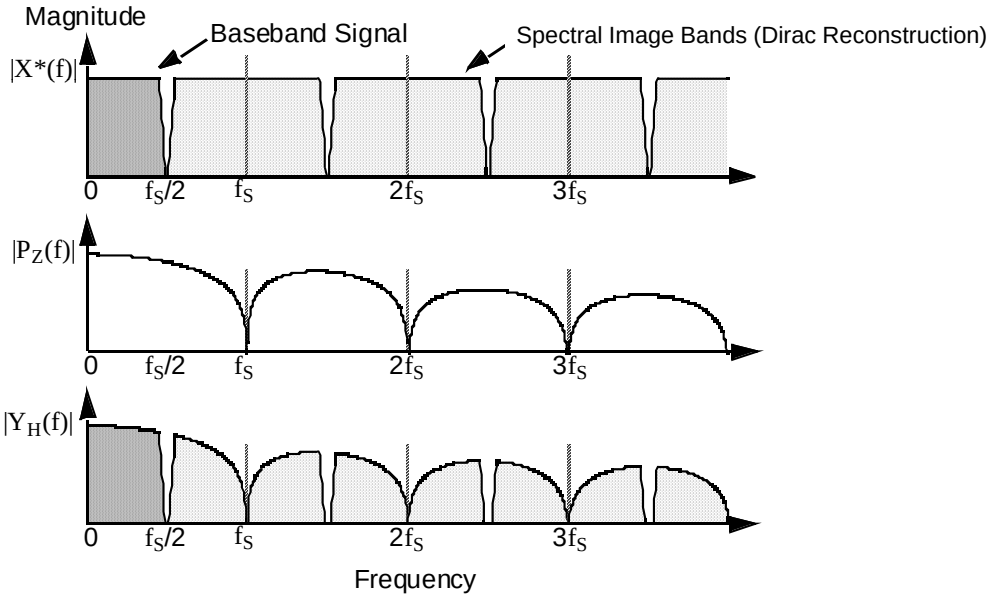
- Digital signal
- Discrete in time
- Discrete in amplitude

- Analog signal
- Continuous in time
- Continuous in amplitude

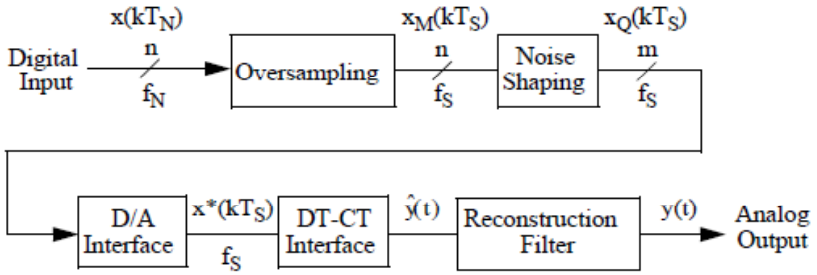
Recap (2)



Frequency Spectra

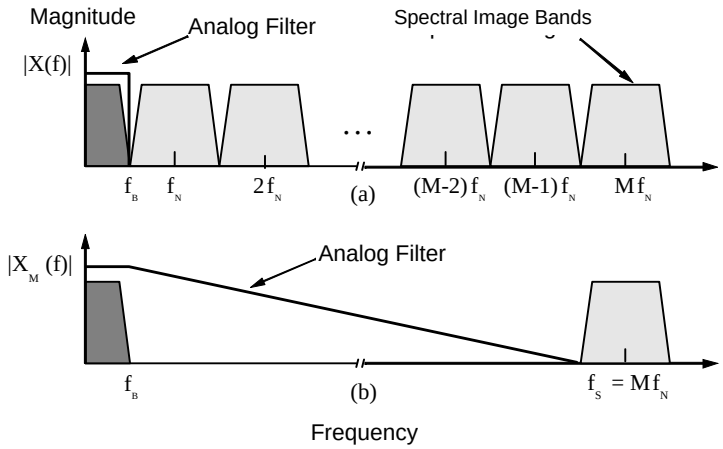


Oversampling D/A Converter



- Oversampling
  - relaxes requirements on reconstruction filter
- Noise shaping
  - Encoding in single bit code (m=1)
  - Relaxes requirements on D/A interface
- BUT … addition of quantization noise, fortunately most of which is out-of-band

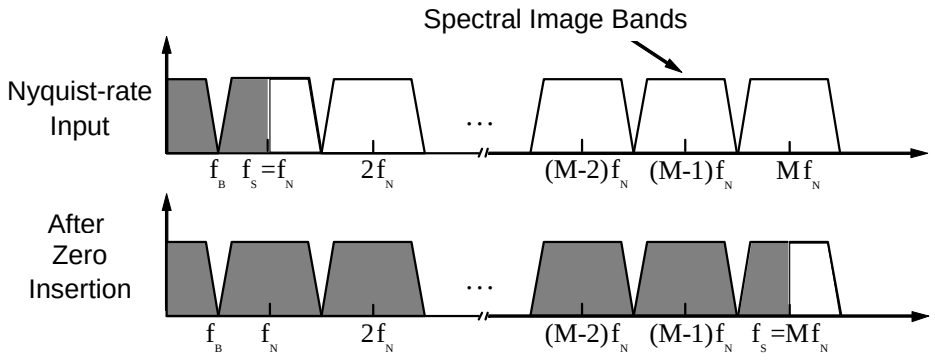
Oversampling



- Oversampling greatly reduces reconstruction filter requirements
- How to create oversampled DAC input from a Nyquist rate signal?

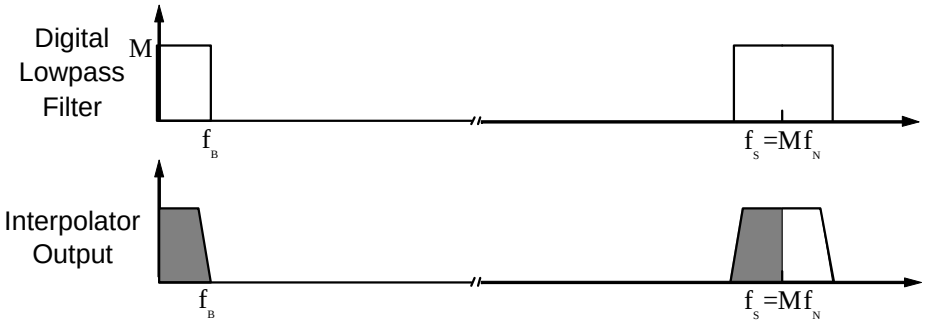
Interpolation (1)

- Can increase the sampling rate of a discrete time signal by a factor of M, by inserting M-1 zero-valued samples between the actual Nyquist rate samples ("zero stuffing")
  - Causes an M-fold periodic repetition of the baseband spectrum

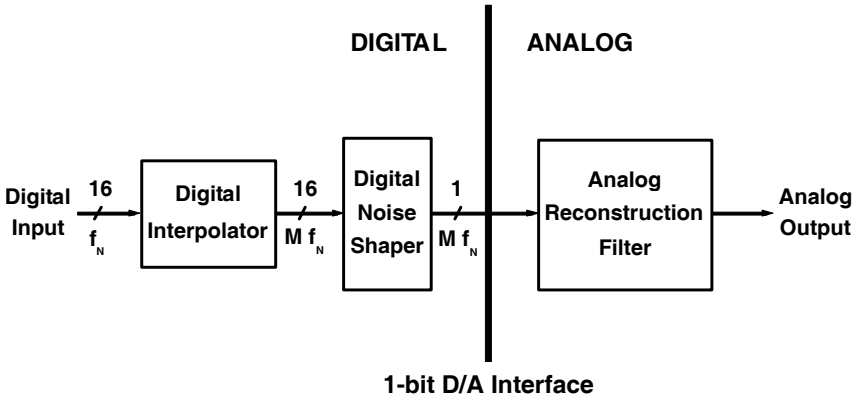


Interpolation (2)

- Why is this a good idea?
- Can remove images and get wide transition band to play with
  - Simple reconstruction filter
  - Possibility of noise shaping
    - Build a high resolution DAC using a low resolution D/A interface

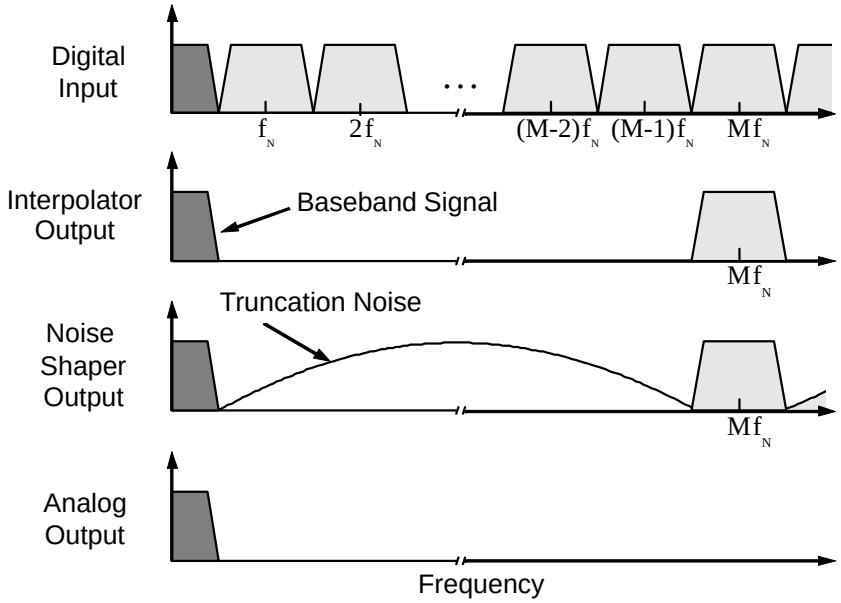


Example

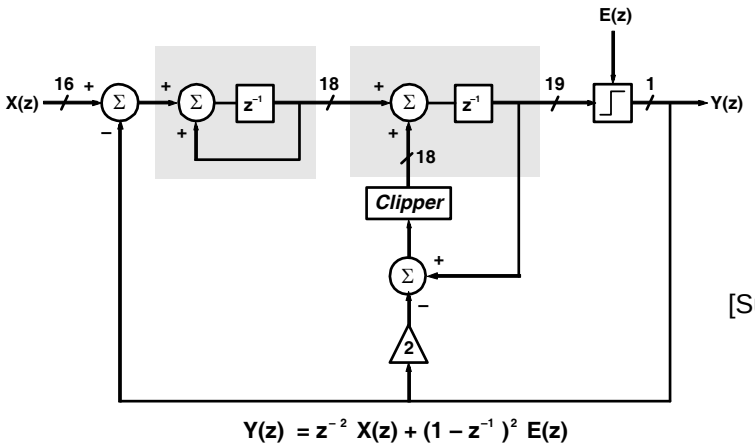


- Digital noise shaper is essentially a digital sigma-delta loop
  - Shapes "truncation noise" that results from truncating 16-bit word to a 1-bit output

Spectra



Example

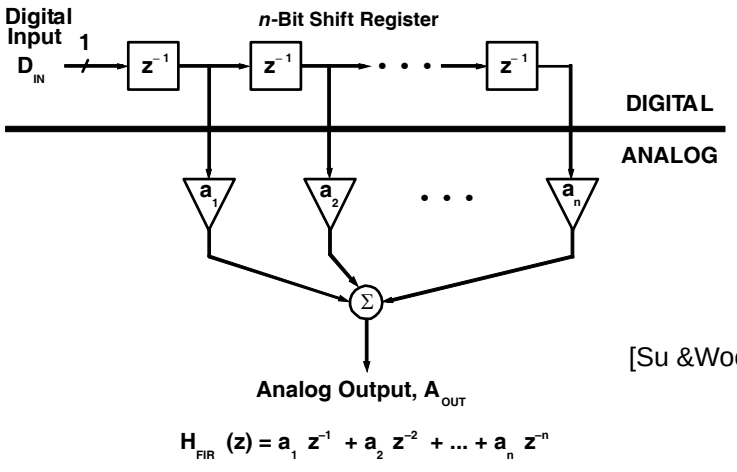


To 1-bit D/A Interface

[Su &Wooley, JSSC 12/94]

- Clipper prevents second integrator from overflowing
  - Digital "wrap around" would cause large errors

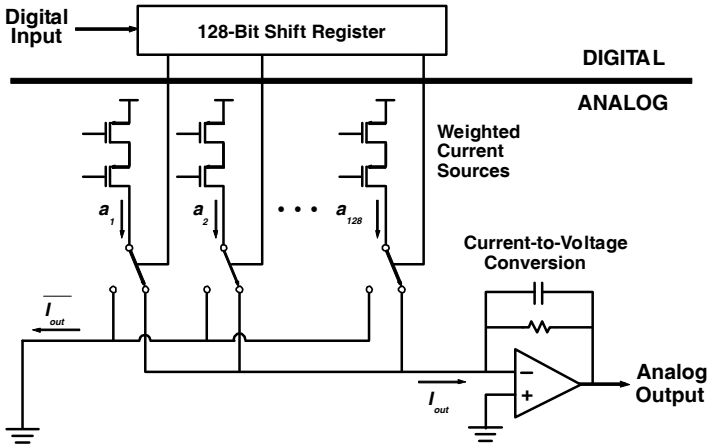
Semi-Digital Reconstruction (1)



[Su &Wooley, JSSC 12/94]

- Attractive alternative to fully analog reconstruction filter
  - Build FIR filter with weighted analog outputs

Semi-Digital Reconstruction (2)

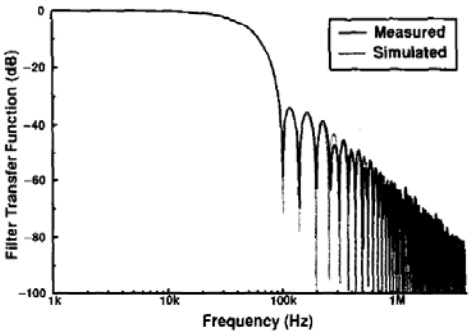


$$A_{OUT}(z) = \underbrace{\left[a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \ldots + a_n z^{-n}\right]}_{H(z)} D_{IN}(z)$$

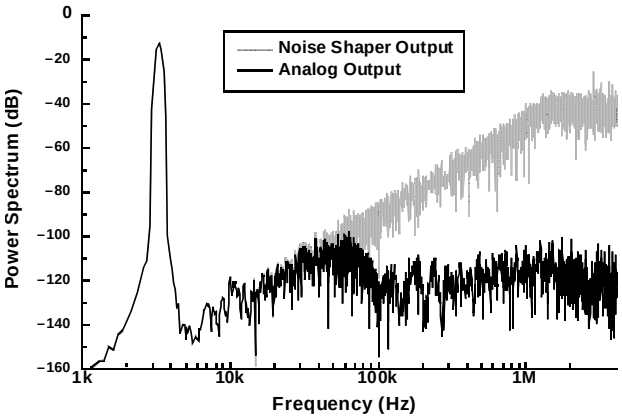
- Linear if H(z) is independent of $D_{IN}(z)$

Measurement Results

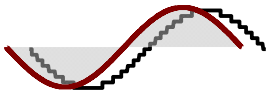
Reconstruction Filter



Output



ADC Figures of Merit
Limits on ADC Power Dissipation



Katelijn Vleugels
Stanford University

ADC Figures of Merit (1)

- Objective
  - Want to compare performance of different ADCs
- Can use FOM to combine several performance metrics into one single number
- What are reasonable FOMs for ADCs?
- How can we use and interpret them?
- Trends and limits?

- Reference: B. Murmann, “Limits on ADC Power Dissipation,” in *Analog Circuit Design,* by M. Steyaert, A.H.M Roermund, J.H. van Huijsing (eds.), Springer, 2006.

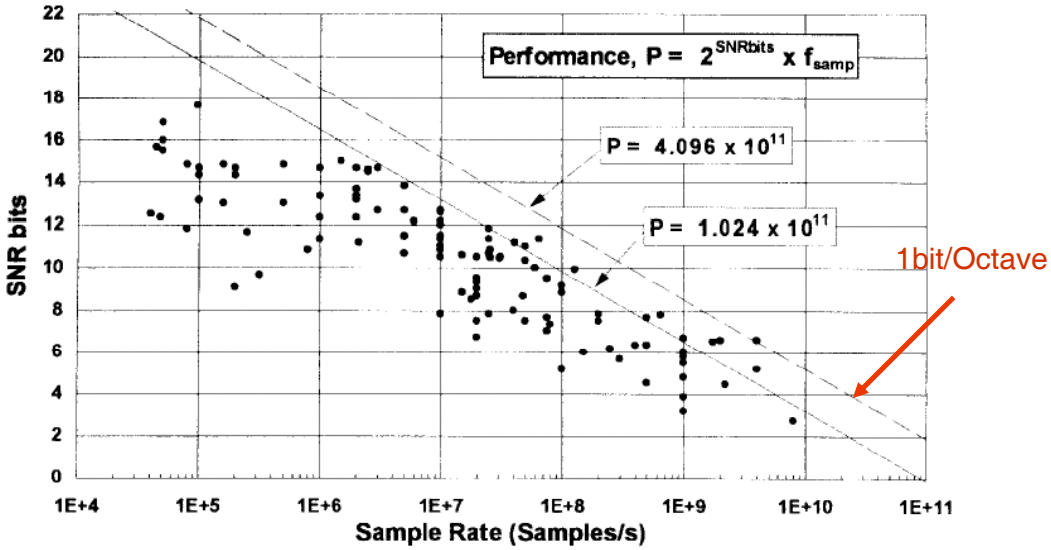
ADC Figures of Merit (2)

$$FOM_1 = f_s \cdot 2^{ENOB}$$

[R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, April 1999]

- This FOM suggests that adding a bit to an ADC is just as hard as doubling its bandwidth
- Is this a good assumption?

Survey Data



[Walden, "Analog-to-digital converter survey and analysis," *IEEE J. Selected Areas Comm.*, April 1999]

ADC Figures of Merit (3)

$$FOM_2 = \frac{f_s \cdot 2^{ENOB}}{Power}$$

[R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, April 1999]

- Sometimes inverse of this metric is used
- In typical circuits power ~ speed
  - FOM_2 captures this tradeoff correctly
- How about power vs. ENOB?
  - One additional bit = 2x in power?

ADC Figures of Merit (4)

- In a circuit that is limited by thermal noise, each additional bit in resolution means...
  - 6dB SNR, 4x less noise power, 4x bigger C
  - Power ~ G_m ~ C increases 4x
- Even worse: Flash ADC
  - Extra bit means 2x number of comparators
  - Each of them needs double precision
  - Transistor area 4x, Current 4x to maintain current density
  - Net result: Power increases 8x

ADC Figures of Merit (5)

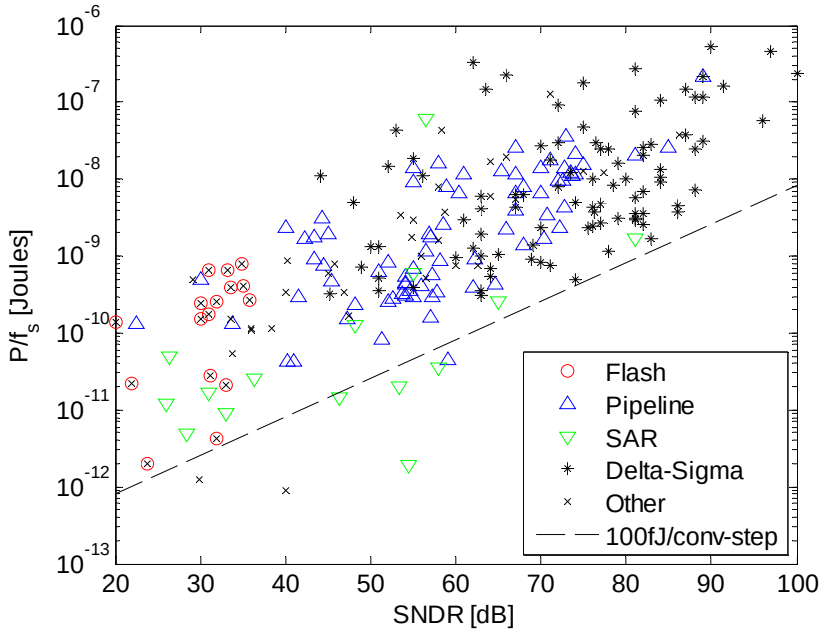
- FOM_2 in inappropriate for comparing ADCs that are limited by matching or thermal noise
  - Still the most widely used FOM in publications...
- "Tends to work" because not all power in an ADC is noise limited
  - E.g. Digital power, biasing circuits, etc.
- To better capture the case of noise limited circuits, one could use 2^{2ENOB} in the numerator of FOM2...
  - But how about other (non-noise limited) circuits?
- My suggestion
  - Avoid using a FOM that assumes a fixed relationship between ENOB and power

ADC Figures of Merit (6)

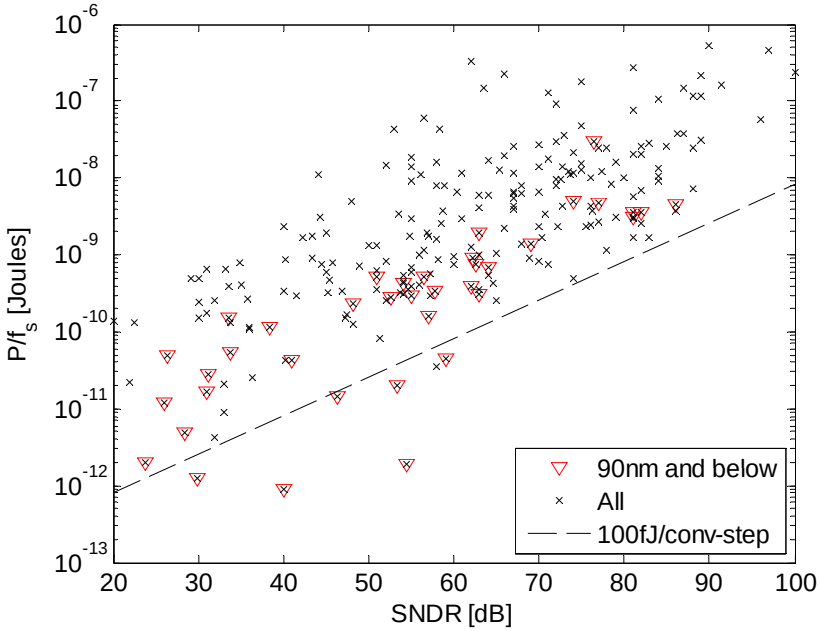
$$FOM_3 = \frac{Power}{2 \cdot \text{Conversion Bandwidth}} = \text{"Energy per Nyquist Sample"}$$

- Compare only power of ADCs with approximately same SNR or SNDR (ENOB)
- Useful numbers (~state-of-the-art):
  - 10b (~9 ENOB) ADCs: 0.25...1 mW/MHz
  - 12b (~11 ENOB) ADCs: 2...6 mW/MHz

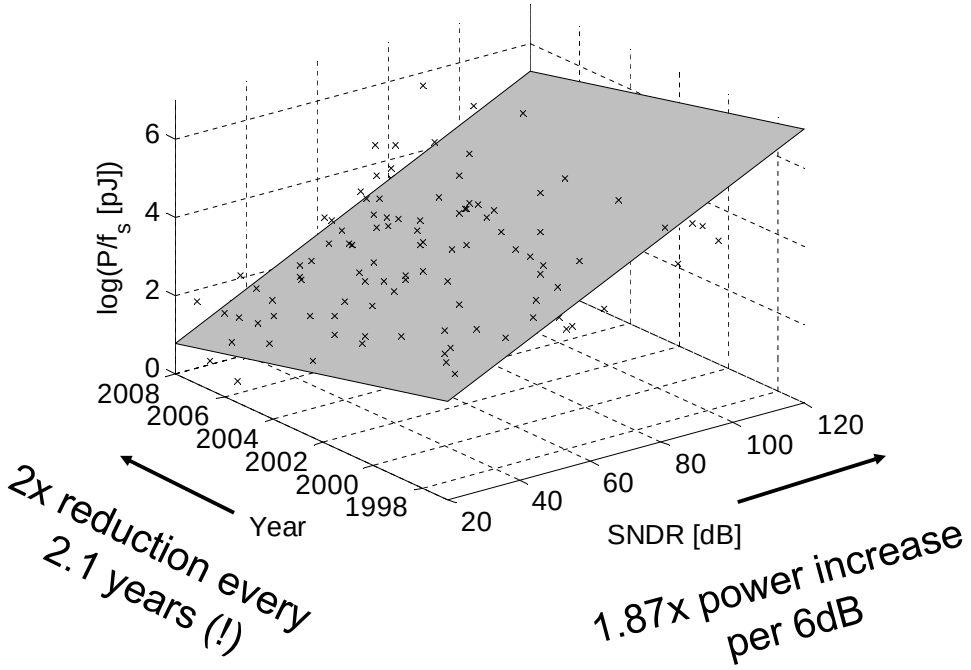
FOM3 (ISSCC & VLSI 1997-2008)



Power Dissipation in Sub-100nm CMOS



Power Dissipation Trend



Fundamental Limits

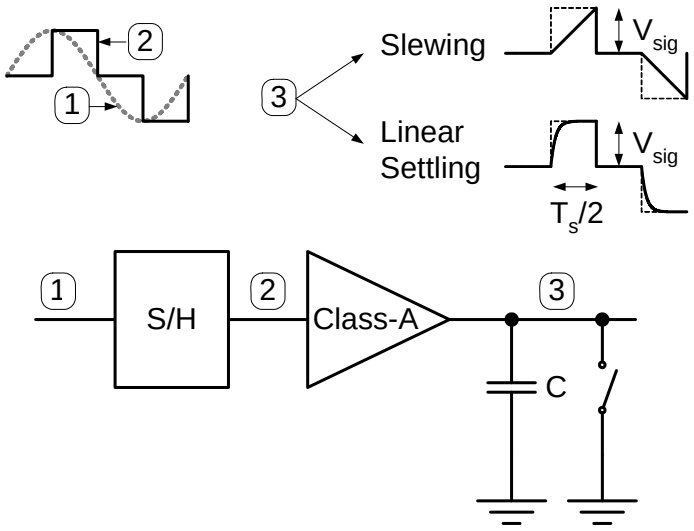
- Fundamental power limit for a class-B amplifier driving a single capacitor [Vittoz, ISCAS 1990]

$$P = 8 \cdot f_{sig} \cdot CV_{sig}{}^2 \qquad V_n^2 = \frac{k_B T}{C} \qquad SNR = \frac{0.5 \times V_{sig}^2}{V_n^2}$$

$$\boxed{\therefore P = 8 k_B T \cdot SNR \cdot f_{sig}}$$

- Class-A power limit is π times higher

Switched Capacitor Circuits



Case 1: 100% Slewing

$$I_{bias} = C \cdot \frac{dV}{dt} = C \cdot \frac{V_{sig}}{T_s / 2} = 4 \cdot C \cdot V_{sig} \cdot f_{sig}$$

$$P = 2 \cdot V_{sig} \cdot I_{bias} \qquad SNR = \frac{0.5 \times V_{sig}^2}{k_B T / C}$$

$$\boxed{\therefore P = 16 k_B T \cdot SNR \cdot f_{sig}}$$

Case 2: 100% Linear Settling

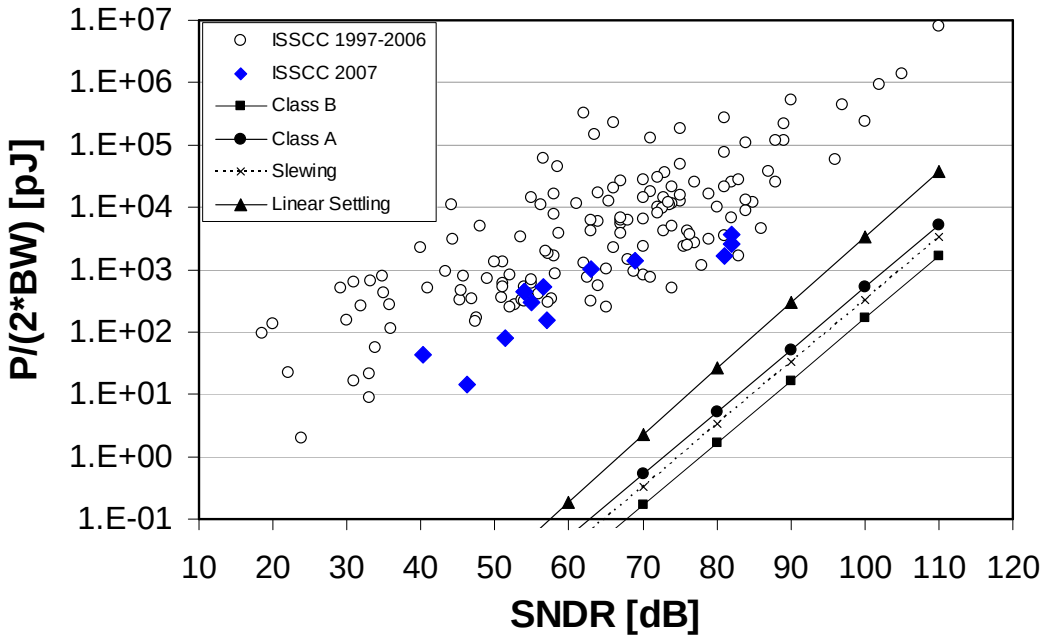
$$I_{bias} = C \cdot \left.\frac{dV}{dt}\right|_{max} = C \cdot \left.\frac{d}{dt}\right|_{max} \left[V_{sig}\left(1 - e^{-t/\tau}\right)\right] = C \cdot \frac{V_{sig}}{\tau}$$

Number of settling time constants: $N = \frac{T_s / 2}{\tau}$

$$\therefore P = 16 \cdot N \cdot k_B T \cdot SNR \cdot f_{sig}$$

- Much worse
  - E.g. N=6.9 for settling to 0.1% precision

Limit Lines



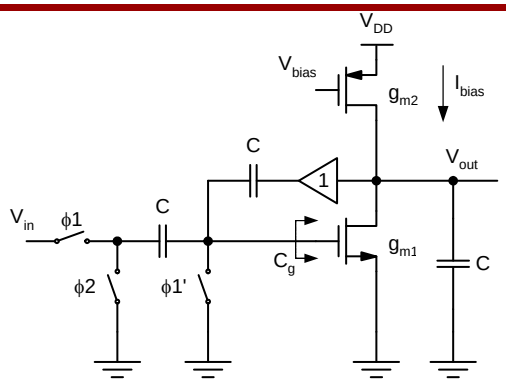
Discussion

- Orders of magnitude away from limits
- Slope of limit lines is much steeper than fit to experimental data
- What contributes to these large gaps?
  - Must keep in mind that ADCs are not just a single capacitor circuit…
- The following analysis factors in practical considerations
  - Not fundamental, but somewhat unavoidable in today's implementations

Design Space Partitioning

- High SNR
  - Complexity ~1 (e.g. first integrator in sigma-delta ADC)
  - Limited by thermal noise
- Moderate SNR
  - Complexity ~Bits (e.g. pipelined ADC)
  - Partly limited by thermal noise
- Low SNR
  - Complexity ~2^{Bits} (e.g. flash ADC)
  - Limited by matching, quantization noise

High SNR SC-Stage (1)



- Considerations
  - Noise is multiple of $k_B T/C$ (n_f)
  - Swing is only a fraction of V_{DD} (α)
  - Feedback factor (β)
  - g_m/I_D is upper bounded if slewing must be avoided

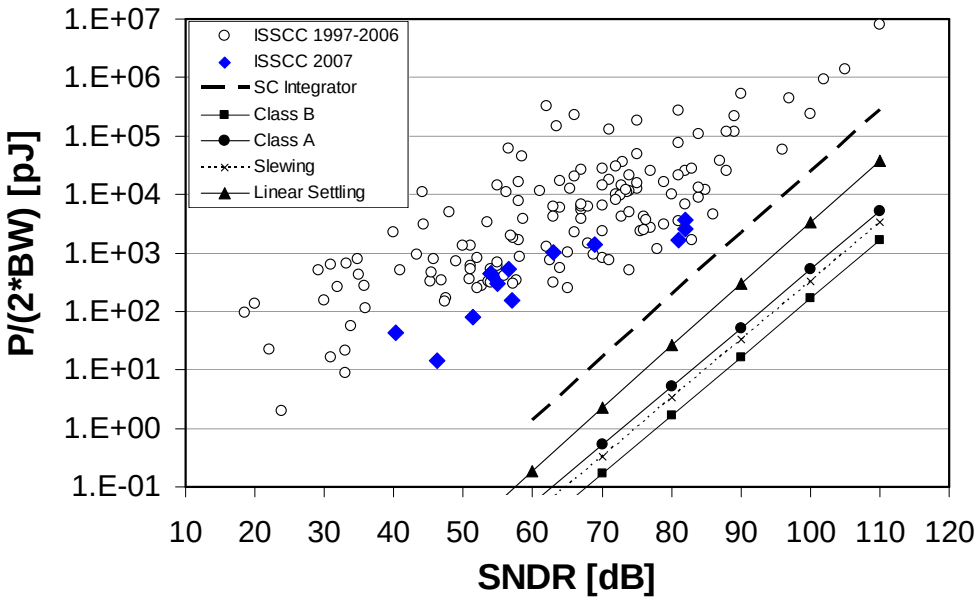
High SNR SC-Stage (2)

To avoid slewing: $$\frac{g_{m1}}{I_{bias}} \leq \frac{1}{\beta \cdot V_{sig}}$$

$$\therefore P = 16 \cdot N \cdot n_f \cdot \frac{1}{\alpha} \cdot k_B T \cdot SNR \cdot f_{sig} \cdot max\left(1, \frac{1}{\frac{g_{m1}}{I_{bias}} \cdot \beta \cdot V_{sig}}\right)$$

- Graph on following slide shows result assuming
  - n_f=5, α=2/3, β=0.5, onset of slewing

High SNR SC-Stage (3)

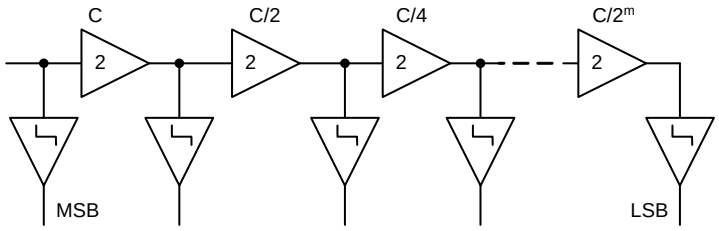


- Close to experimental data at high SNDR!

Medium SNR

- Consider two cases
- Pipeline ADC using SC stages
  - Partially limited by thermal noise
- Continuous time G_m-C integrator
  - Limited by distortion

Pipeline ADC



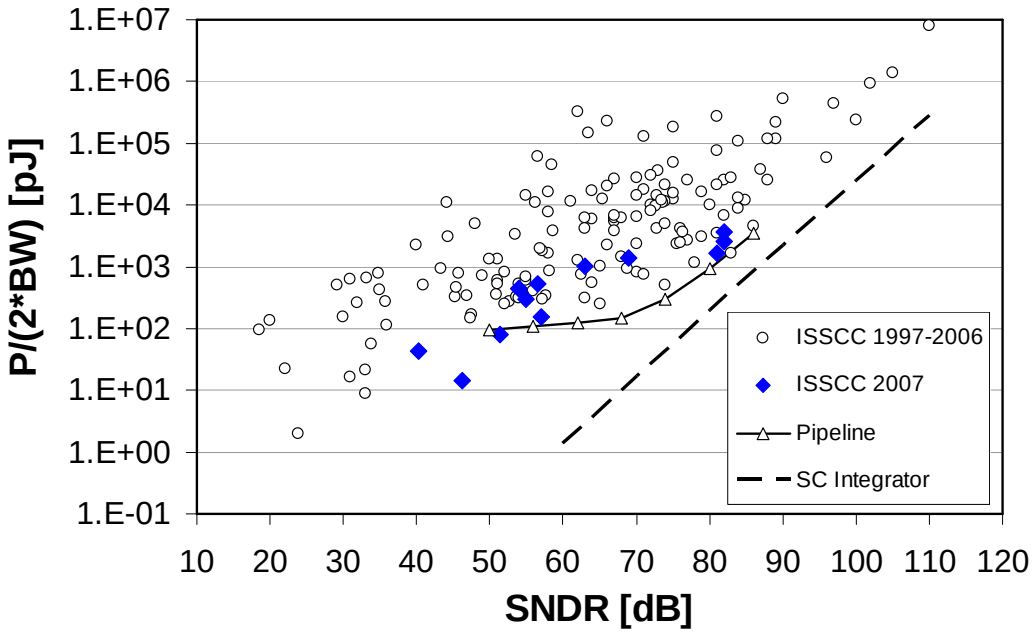
- Theoretical near optimum power scaling
  - Scale capacitance by gain of preceding stage
  - Stage 1 consumes half of total power
  - Adding one bit means power goes up 4x
- Caveat
  - Usually impractical to scale capacitors down to $C/2^m$

Stage Scaling Example

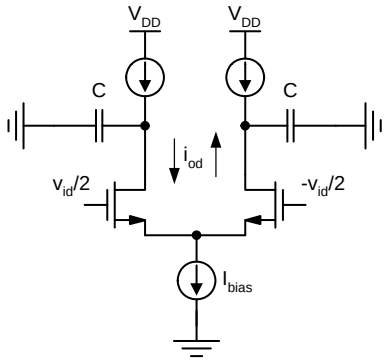
Number of Amplifiers	**13**	**12**	**11**	**10**
Stage Capacitances	1	1/4	1/16	1/64
	1/2	1/8	1/32	1/128
	1/4	1/16	1/64	*1/128*
	1/8	1/32	1/128	*1/128*
	1/16	1/64	*1/128*	*1/128*
	1/32	1/128	*1/128*	*1/128*
	1/64	*1/128*	*1/128*	*1/128*
	1/128	*1/128*	*1/128*	
	1/128	*1/128*		
	1/128			
ΣC	2.03	0.54	0.17	0.086
C_{single}	1/2	1/8	1/32	1/128
Relative Power Pipeline/Single SC Stage ($\Sigma C/C_{single}$)	4.06	4.32	5.44	11.01

- Example is simplistic, but in line with state-of-the art
  - 10bits ~0.5mW/MSample/s, 12bits ~2mW/MSample/s

Pipeline ADC Limit Line



G_m-C Integrator



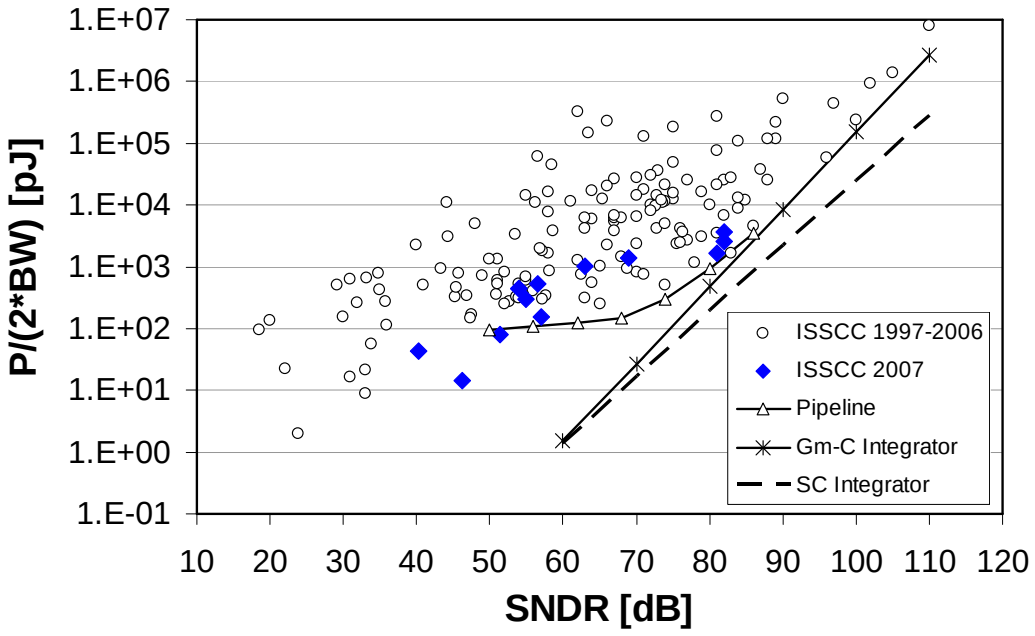
$$IM_3 \cong \frac{3}{32}\left(\frac{v_{id,max}}{V_{ov}}\right)^2$$

$$\eta_{cur} = \frac{i_{od,max}}{I_{bias}}$$

$$\eta_{cur} \cong \frac{v_{id,max}}{V_{ov}} = \sqrt{\frac{32}{3} IM_3}$$

- Only a small fraction of bias current can be steered into load
  - E.g. IM_3=60dB, η_{cur}=10%

Gm-C Limit Line



Low SNR

- Power of matching limited class-B circuit [Kinget, CICC 1996]

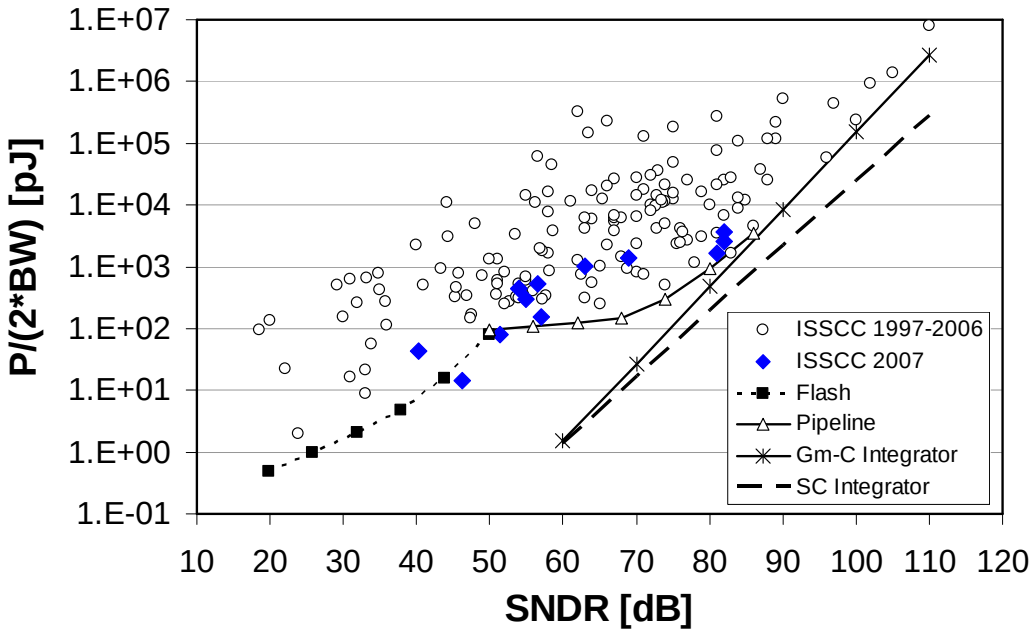
$$P = 24 \cdot C_{ox} \cdot A_{VT}^2 \cdot f_{sig} \cdot \left(\frac{V_{sig,rms}}{3 \cdot \sigma_{Vos}} \right)^2$$

- Refined result for flash ADC, assuming
  - Class-A, 1/2 LSB matching with 3σ-confidence, 2^B components, additional E_{dyn} per clock cycle, partial supply usage (α)

$$P = \left(12\pi \cdot \frac{1}{\alpha} \cdot C_{ox} \cdot A_{VT}^2 \cdot 2^{3B} + 2 \cdot E_{dyn} \cdot 2^B \right) \cdot f_{sig}$$

- Example: α=2/3, C_{ox}=15fF/μm², A_{Vt}=3mV·μm, E_{dyn}=60fJ (~10gates in 0.13μm CMOS)

End Result



Discussion

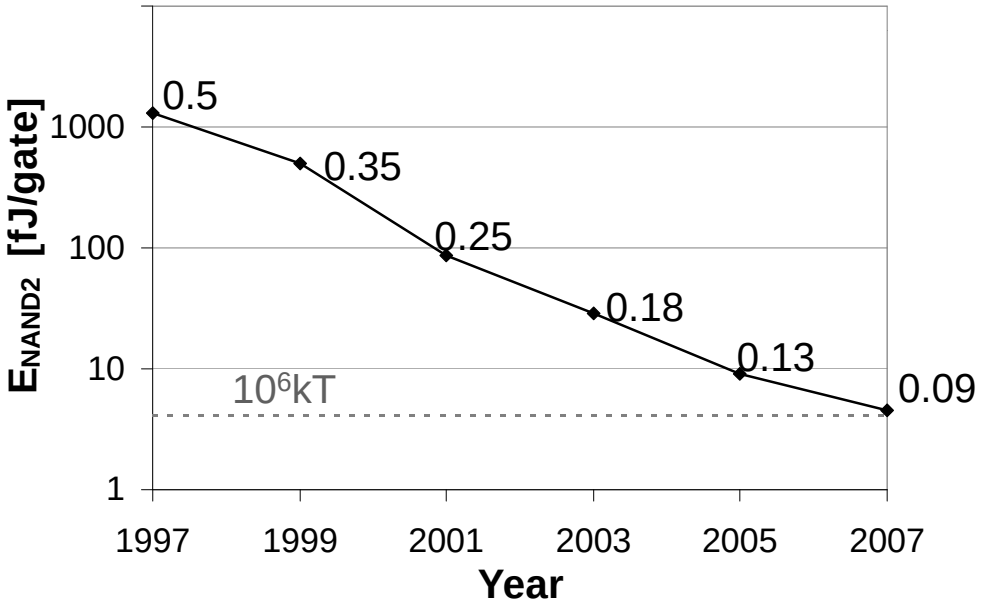
- Shown results include only minor assumptions about technology
- Scaling brings some good, some bad news offsetting each other
  - – Lower V_{DD}, lower V_{swing}/V_{DD}, …
  - + Lower E_{dyn}, higher f_t enables moderate/weak inversion operation with high g_m/I_D,…
- Limit lines won't move much, unless someone hands us a new disruptive technology

Future Opportunities

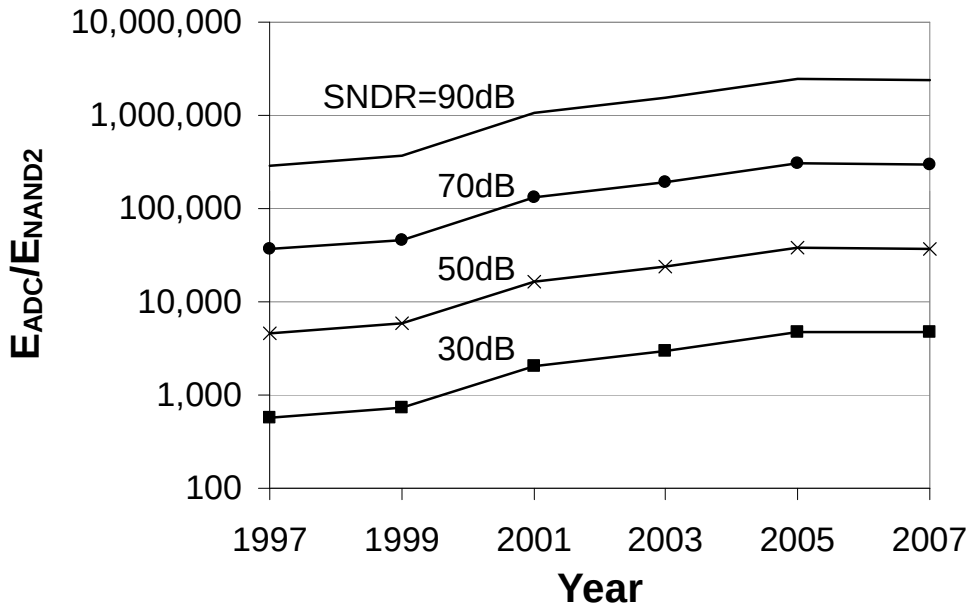
- More intelligent ADCs
  - Improved average power dissipation by adapting to instantaneous speed/resolution requirements
- "Minimalistic" ADCs using significantly simpler circuits
  - Digital compensation of resulting non-idealities
  - Digital postprocessing is (within limits) "free" in terms of area and energy

Digital Logic Energy Trend

Mainstream ADC technologies, standard logic library data



ADC/Digital Logic Energy Ratio



Energy Ratio in 2007

- Interpretation for digitally enhanced ADCs (energy centric)

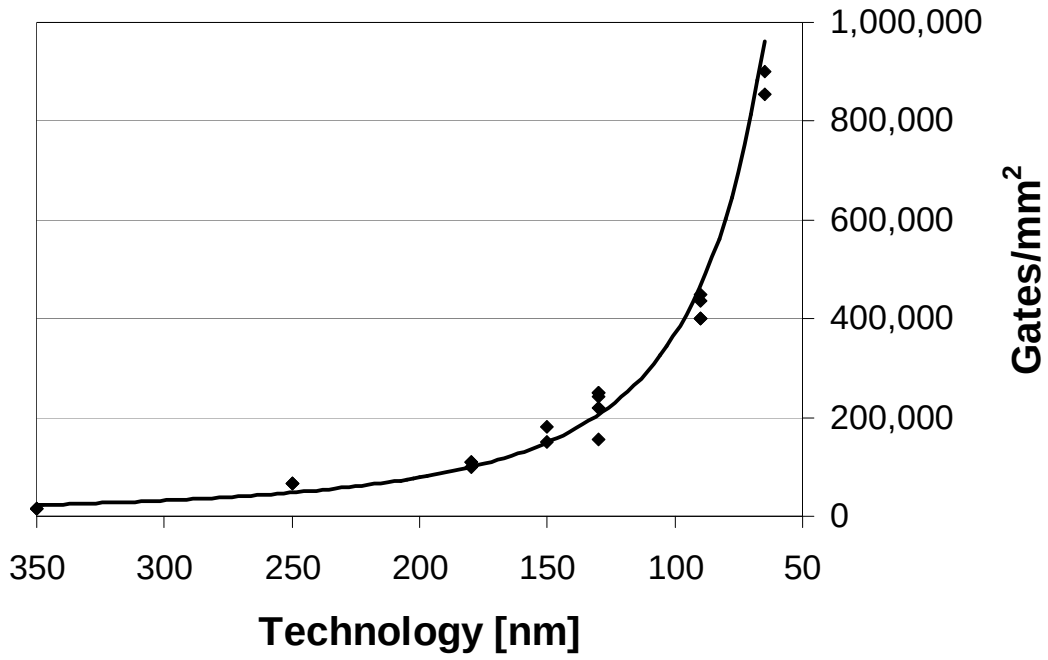
SNDR	E_{ADC}/E_{NAND2}
30	4,679
50	37,432
70	299,479
90	2,396,045

⇨ Additional digital processing is costly!

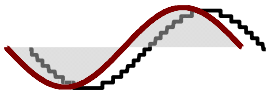
⇨ Several tens of thousand gates are "free"

⇨ Use as many gates as you can fit…

Digital Logic Gate Density Trend



Data Converter Testing

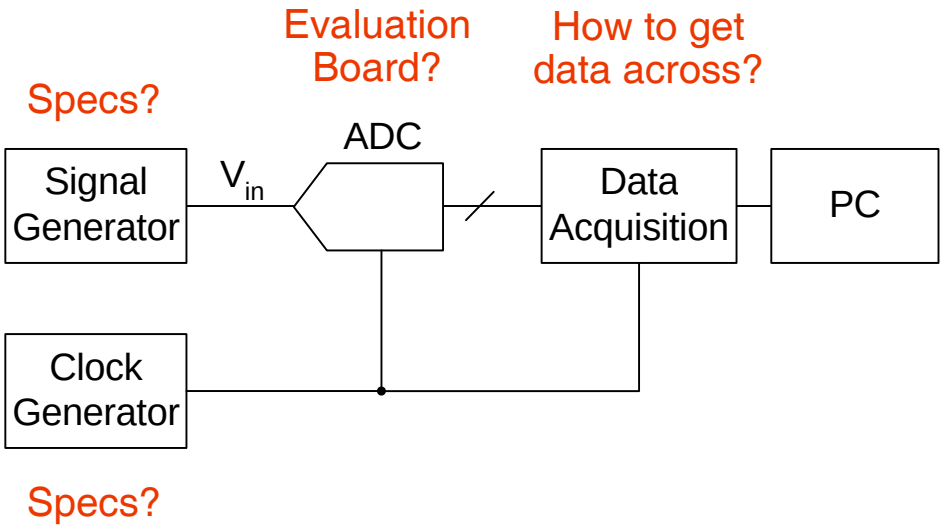


Katelijn Vleugels
Stanford University

Just Got Silicon Back...

- Now what ?
- Practical aspects of converter testing
- Equipment requirements
- Pitfalls

ADC Test Setup



High-Performance ADC Specs

Resolution	14 bits
Conversion Rate	75 MSPS
Input Range	2 V_{pp} differential
SNR @ Nyquist	73 dB
SFDR @ Nyquist	88 dB
DNL	0.6 LSB
INL	2.0 LSB

[W. Yang et al., "A 3-V 340-mW 14-b 75-Msample/s CMOS ADC with 85-dB SFDR at Nyquist input," *IEEE J. of Solid-State Circuits,* Dec. 2001]

- Your converter will perform even better...
- Testing a high performance converter may be just as challenging as designing it!
- Key to success is to be aware of test setup and equipment limitations

Signal Source

- Want: SFDR>85dB @ $f_{in}=f_s/2=37.5$MHz
- Let's see, how about the "value priced" signal generator we have in the lab...

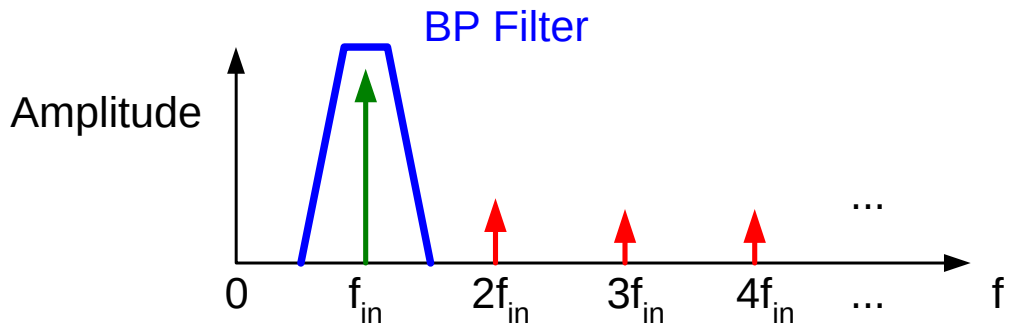
- f=0...15MHz
- Harmonic distortion (f>1MHz): -35dBc
- Need something better...

A Better Signal Source

- OK, now we've spent about $40k, this should work now... (?)

- f=100kHz...3GHz
- Harmonic distortion (f>1MHz): -30dBc !
- No way to produce the sine wave we need without a filter!

Filtering Out Harmonics



- Given HD=-30dBc, we need a stopband rejection > 60dB to get SFDR>90dB

Available Filters

Elliptical Function Bandpass Filters 1kHz to 20MHz

www.tte.com, or
www.allenavionics.com

Stopband to Passband Bandwidth Ratios

Series Number	BWR	*Stopband Attenuation
Q34	4.0:1	-40dBc
Q40	4.0:1	-40dBc
Q36	10.0:1	-60dBc
Q54	2.5:1	-40dBc
Q70	3.5:1	-60dBc
Q56	3.5:1	-60dBc

- Want to test at many frequencies -> Need to have many different filters!

Tunable Filter

www.klmicrowave.com

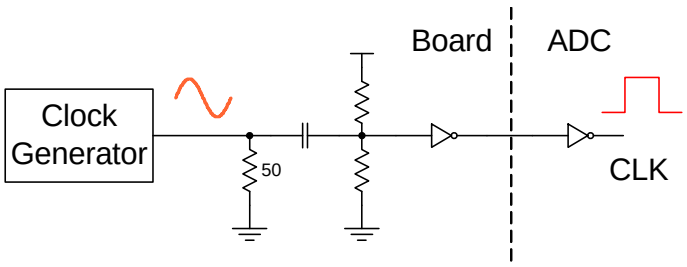
K&L Model	Frequency Range (MHz)	Passband Insertion Loss	Length Inch/mm	Width Inch/mm	Height Inch/mm
5BT-30/76-5-N/N	30-76	1.3 dB Max	9.80/249	5.38/137	2.75/50
5BT-63/125-5-N/N	63-125	1.3 dB Max	9.80/249	5.38/137	2.75/50
5BT-125/250-5-N/N	125-250	1.3 dB Max	9.80/249	5.38/137	2.75/50
5BT-250/500-5-N/N	250-500	1.0 dB Max	9.80/249	5.38/137	2.75/50
5BT-375/750-5-N/N	375-750	1.0 dB Max	9.80/249	5.38/137	2.75/50
5BT-500/1000-5-N/N	500-1000	1.0 dB Max	9.80/249	5.38/137	2.75/50
5BT-750/1500-5-N/N	750-1500	1.0 dB Max	9.80/249	5.38/137	2.75/50
5BT-1000/2000-5-N/N	1000-2000	1.0 dB Max	7.38/187	2.88/73	2.75/50
5BT-1200/2600-5-N/N	1200-2600	1.0 dB Max	7.38/187	2.88/73	2.75/50

Filter Distortion

- Beware: The filters themselves also introduce distortion
- Distortion is usually not specified, need to call manufacturer
- Often guaranteed: HD<-85dBc,
- Don't trust your filters blindly...

Clock Generator

- OK, may be for the clock a "value-priced" signal generator will suffice...
- No! The clock signal controls sampling instants – which we assumed to be precisely equidistant in time (period T)
  - See Lecture for a dscussion of aperture uncertainty
- Typically use sine wave and "square up" with inverter chain
  - Jiiter requirements ⇔ sine wave specs



Phase Noise and Jitter

$$\tau = \sqrt{\frac{1}{f_0}} \cdot \frac{\Delta f}{f_0} \cdot 10^{L(\Delta f)\cdot[Hz]/20} \cdot \left[\frac{1}{\sqrt{Hz}}\right]$$

"Cycle to cycle jitter"

Phase Noise at offset Δf from "carrier"

[Hajimiri, *The Design of Low Noise Oscillators, p.147*, Kluwer 1999]

- Can use the above equation to get a (very rough) jitter estimate from phase noise spectrum
- "Value Priced" Signal Generator:
  - L(30kHz)=-55dBc/Hz -> $\tau(f_0=15MHz)$ = 230ps rms
- "$40k" Signal Generator:
  - L(30kHz)=-122dBc/Hz -> $\tau(f_0=15MHz)$ = 0.1ps rms → OK!

More on Jitter

- Once we have a good enough generator, other circuit and test setup related issues may determine jitter
- Usually, clock jitter in the single-digit picosecond range can be prevented by appropriate design techniques
  - Separate supplies
  - Separate analog and digital clocks
  - Short inverter chains between clock source and destination
- Few, if any, other analog-to-digital conversion non-idealities have the same symptoms as sampling jitter
  - RMS noise proportional to input frequency
  - RMS noise proportional to input amplitude
- So, if sampling clock jitter is limiting your dynamic range, it's easy to tell, but may be difficult to fix...

Jitter Estimation

- Reference
  - D.M. Hummels, W. Ahmed, W., F.H. Irons, "Measurement of random sample time jitter for ADCs," *Proc. ISCAS*, pp.708-711, May 1995.

$$x(t) = A\cos(\omega_0 t + \theta) \quad (3)$$

$$y_k = A\cos(\omega_0 kT_s + \theta) + g(x(t))|_{t=kT_s} - A\omega_0\Delta_k \sin(\omega_0 kT_s + \theta) + n_k \quad (4)$$

After removal of harmonics:

$$e_k = -A\omega_0\Delta_k \sin(\omega_0 kT_s + \theta) + n_k \quad (5)$$

Spectrum of squared sequence contains a tone proportional to jitter:

$$\begin{aligned} E\{e_k^2\} &= E\{A^2\omega_0^2\Delta_k^2 \sin^2(\omega_0 kT_s + \theta) + n_k^2\} \\ &= E\left\{\left(\frac{A^2\omega_0^2\Delta_k^2}{2} + n_k^2\right) - \frac{A^2\omega_0^2\Delta_k^2}{2}\cos(2\omega_0 kT_s + 2\theta)\right\} \quad (6) \\ &= \left(\frac{A^2\omega_0^2\sigma_\Delta^2}{2} + \sigma_n^2\right) - \frac{A^2\omega_0^2\sigma_\Delta^2}{2}\cos(2\omega_0 kT_s + 2\theta) \quad (7) \end{aligned}$$

Evaluation Board

- Planning begins with converter pin-out
  - Uhps, my clock pin is right next to a digital output...
- Not "black magic", but weeks of design time and "thinking"
- Key aspects
  - Supply/ground routing
  - Bypass capacitors
  - Coupling between signals
- Good idea to look at ADC vendor datasheets for example layouts/schematics/application notes

Vendor Eval Bord Layout

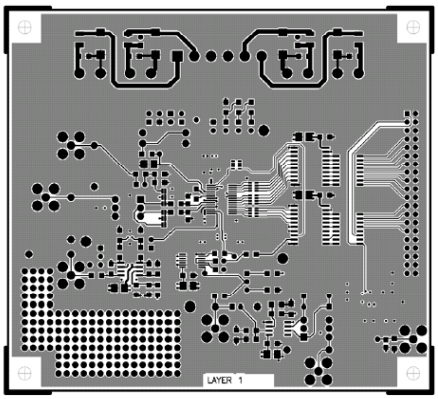


Figure 21. TSSOP Evaluation Board Layout, Primary Side

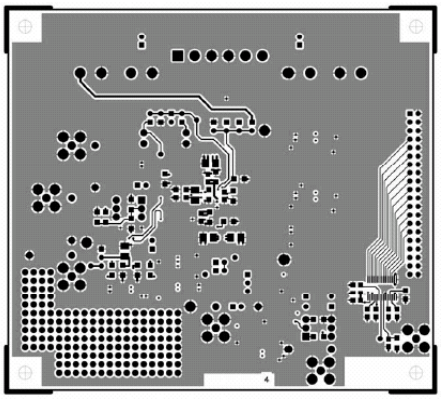


Figure 22. TSSOP Evaluation Board Layout, Secondary Side

[Analog Devices AD9235 Data Sheet]

One Thing to Remember...

- A converter does not just have one "input"
  - Clock
  - Power supply, ground
  - Reference voltage
- For good practices on how to avoid issues see e.g.
  - Analog Devices Application Note 345: "Grounding for Low-and-High-Frequency Circuits"
  - Maxim Application Note 729: "Dynamic Testing of High-Speed ADCs, Part 2"

How to Get the Bits Off Chip?

- "Full swing" CMOS signaling works well for f_{CLK}<100MHz
- But we want to build faster ADCs...
- Alternative to CMOS: LVDS – Low Voltage Differential Signaling
- LVDS vs. CMOS:
  - Higher speed, more power efficient at high speed
  - Two pins/bit!

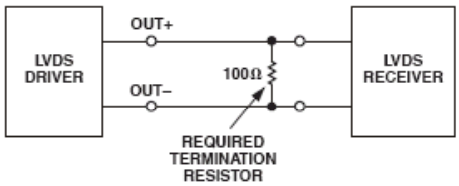


Figure 1. LVDS Output Levels

Analog Devices Application Note 586**:** "LVDS Data Outputs for High Speed ADCs"

LVDS Outputs

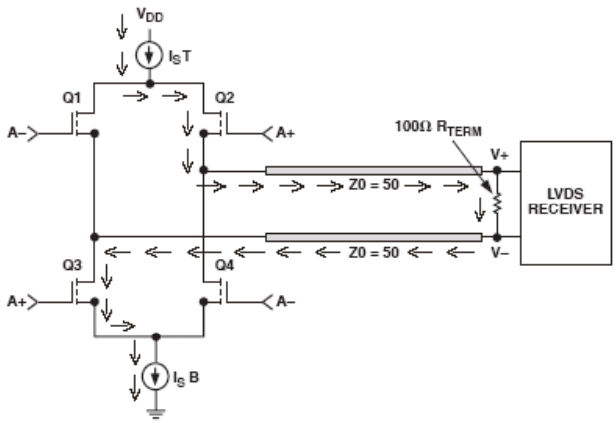
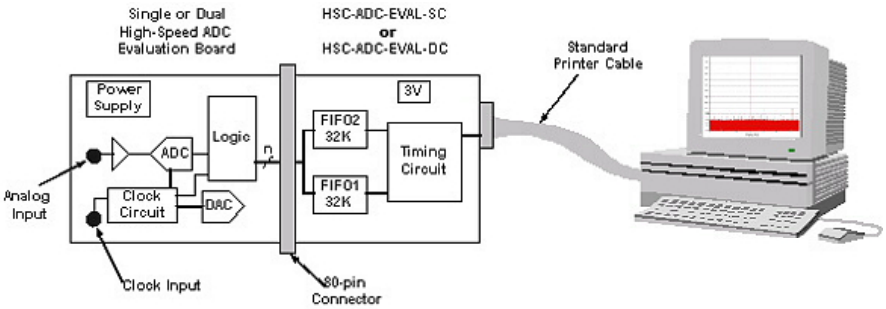


Figure 4. LVDS Output Current

Analog Devices Application Note 586**:** "LVDS Data Outputs for High Speed ADCs"

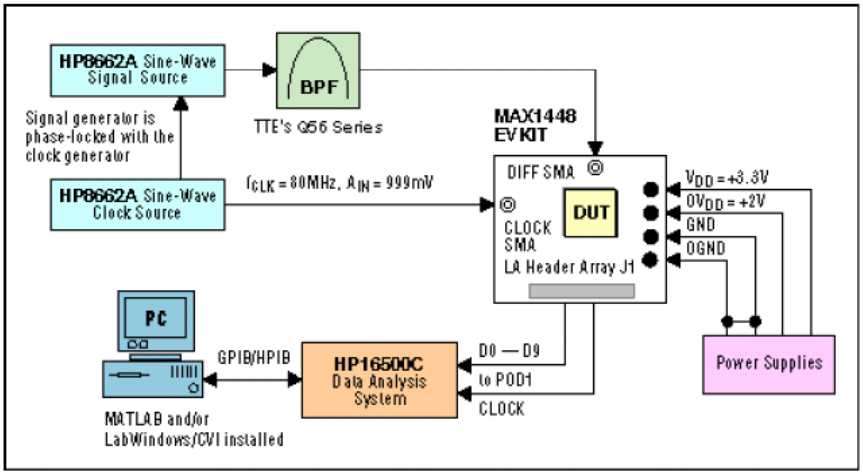
Data Acquisition

- Several options:
  - Logic analyzer with PC interface
  - FIFO board, interface to PC DAQ card
  - Vendor kit, simple interface to printer port:



[Analog Devices, High-Speed ADC FIFO Evaluation Kit]

Complete Setup

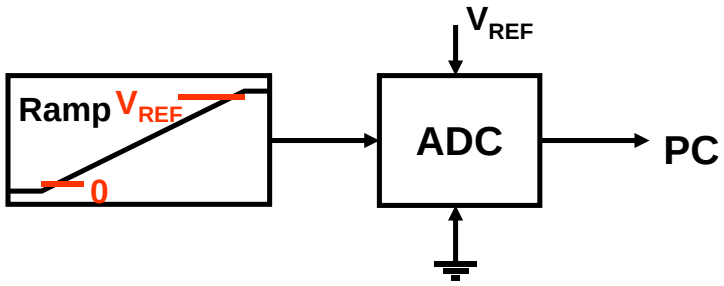


[Maxim Application Note 729**:** "Dynamic Testing of High-Speed ADCs, Part 2]

Post-Processing

- LabView (DAQ Software Toolbox), Matlab
  - Some vendors provide example source code
  - See e.g. Maxim Application Note 1819: "Selecting the Optimum Test Tones and Test Equipment for Successful High-Speed ADC Sine Wave Testing"
- We know how to evaluate spectral metrics
  - How about DNL/INL?
- DAC
  - "Trivial", apply codes and use "a good voltmeter" to measure outputs
- ADC
  - Need to find "decision levels", i.e. input voltages at all code boundaries
  - One way: Adjust voltage source to find exact code transitions
    - "code boundary servo"
  - More elegant: Histogram testing

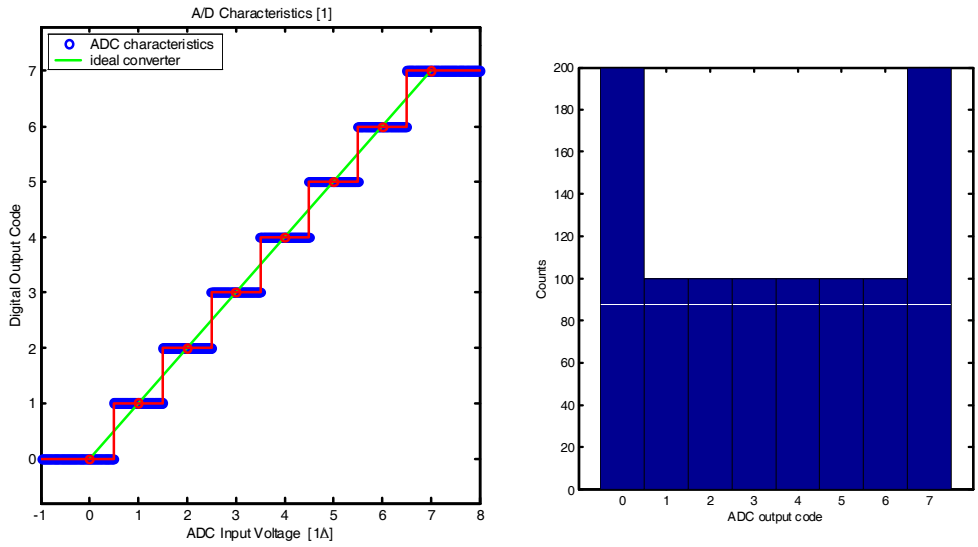
Basic Histogram Test Setup



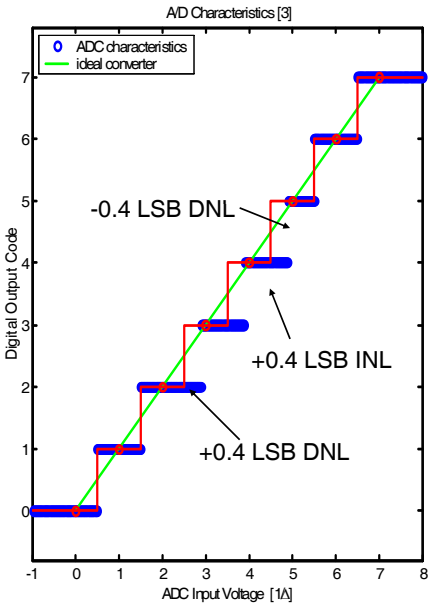
- DNL follows from total number of occurrences of each code
- Ramp speed is adjusted to provide e.g. an average of 100 outputs of each ADC code (for 1/100 LSB resolution)
- Ramps can be quite slow for high resolution ADCs

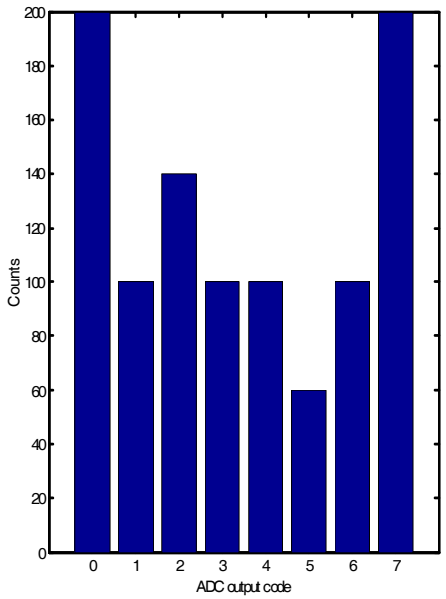
$$\frac{\text{(65,536 codes)(100 conversions/code)}}{\text{100,000 conversions/sec}} = 65.6 \text{ sec}$$

Histogram of Ideal 3 Bit ADC



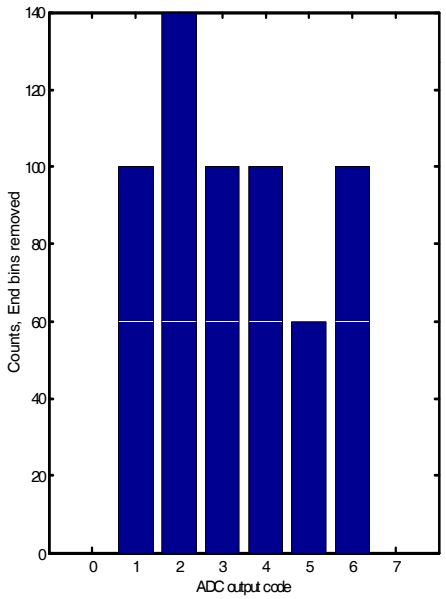
Histogram of Sample 3 Bit ADC





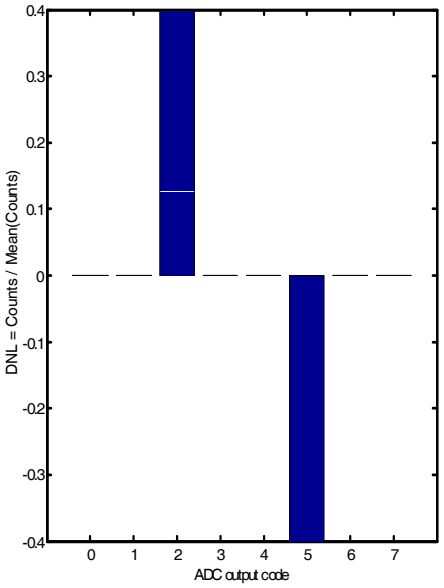
DNL from Histogram (1)

- Step 1
  - Remove “over-range bins” (0 and 7)



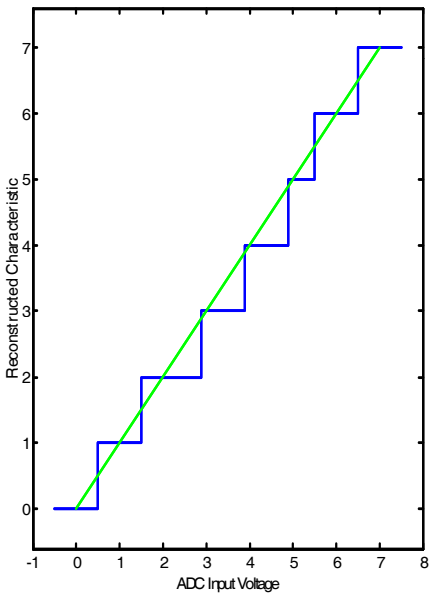
DNL from Histogram (2)

- Step 2
  - Divide by average count
- Step 3
  - Subtract 1
  - Ideal bins have exactly the average count, which corresponds to 1 after normalization
- Result is DNL

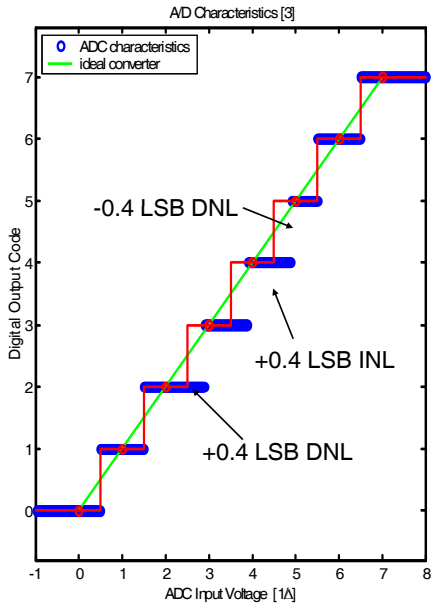


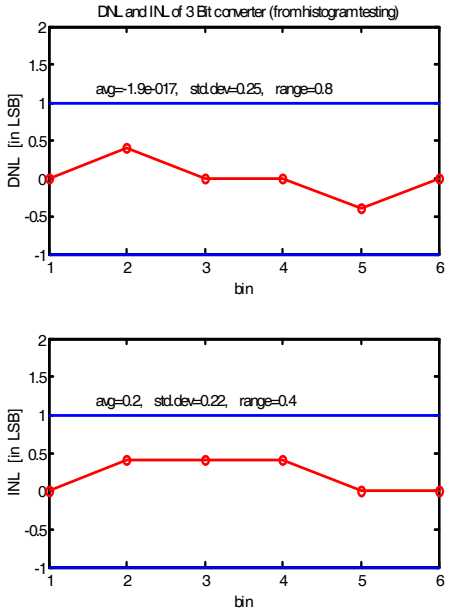
INL from Histogram

- INL is simply running sum of DNL (see HW)
- The DNL information can also be used directly to construct the converter transfer function
  - Simply add up all bin-widths to find transition levels



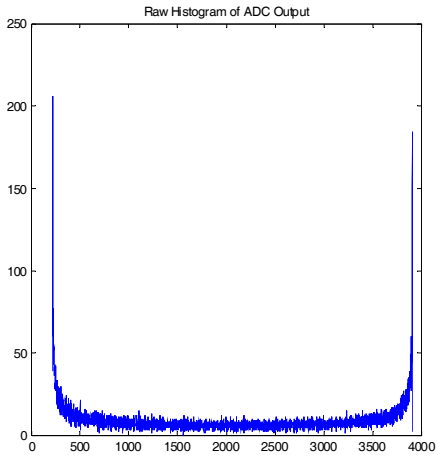
DNL and INL of Sample ADC



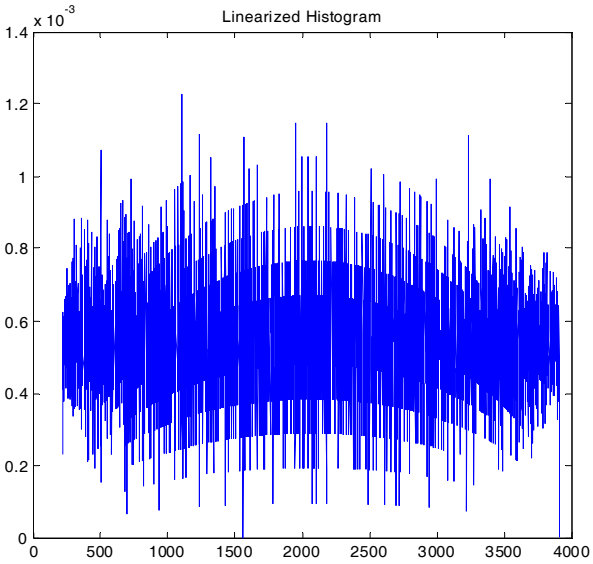


Sinusoidal Inputs

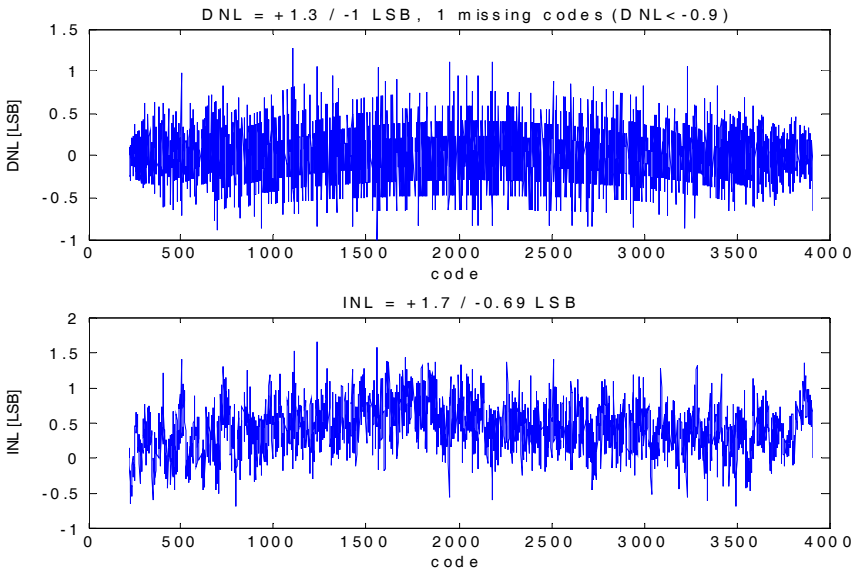
- Precise ramps are hard to generate
- Solution
  - Use sinusoidal test signal
- Problem
  - Ideal histogram is not flat but has “bath-tub shape”



After Correction for Sinusoidal pdf



Resulting DNL and INL



Correction for Sinusoidal pdf

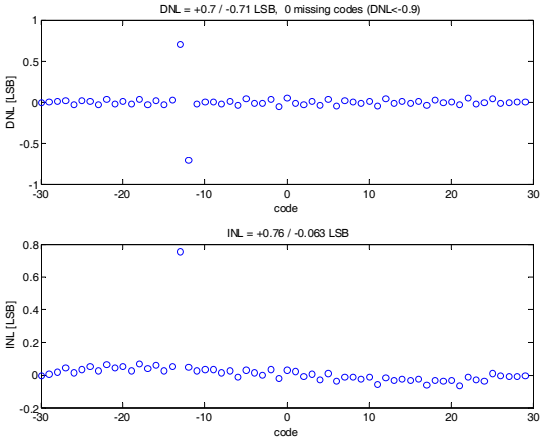
- References
  - M. V. Bossche, J. Schoukens, and J. Renneboog, “Dynamic Testing and Diagnostics of A/D Converters,” IEEE TCAS, Aug. 1986.
  - IEEE Standard 1057
- Is it necessary to know the exact amplitude and offset of the sine wave input?
  - No!
- There exists a great deal of confusion about this in the converter community...

DNL/INL Code

```
function [dnl,inl] = dnl_inl_sin(y);
%DNL_INL_SIN
% dnl and inl ADC output
% input y contains the ADC output
% vector obtained from quantizing a
% sinusoid

% Boris Murmann, Aug 2002
% Bernhard Boser, Sept 2002

% histogram boundaries
minbin=min(y);
maxbin=max(y);

% histogram
h = hist(y, minbin:maxbin);

% cumulative histogram
ch = cumsum(h);

% transition levels
T = -cos(pi*ch/sum(h));

% linearized histogram
hlin = T(2:end) - T(1:end-1);

% truncate at least first and last
% bin, more if input did not clip ADC
trunc=2;
hlin_trunc = hlin(1+trunc:end-trunc);

% calculate lsb size and dnl
lsb= sum(hlin_trunc) / (length(hlin_trunc));
dnl= [0 hlin_trunc/lsb-1];
misscodes = length(find(dnl<-0.9));

% calculate inl
inl= cumsum(dnl);
```

DNL/INL Code Test

```
% converter model
B = 6;              % bits
range = 2^(B-1) - 1;
% thresholds (ideal converter)
th = -range:range;   % ideal thresholds
th(20) = th(20)+0.7; % error

fs = 1e6;
fx = 494e3 + pi;     % try fs/10!
C  = round(100 * 2^B / (fs / fx));

t = 0:1/fs:C/fx;
x = (range+1) * sin(2*pi*fx.*t);
y = adc(x, th) - 2^(B-1);

hist(y, min(y):max(y));
dnl_inl_sin(y);
```

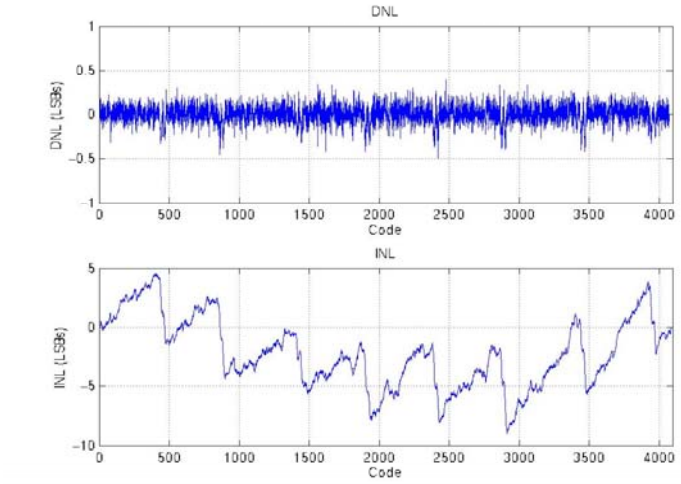


Limitations of Histogram Testing

- The histogram test (as any ADC test, of course) characterizes one particular converter
  - Must test many devices to get valid statistics
- Histogram testing assumes monotonicity
  - E.g. “code flips” will not be detected.
- Dynamic sparkle codes produce only minor DNL/INL errors
  - E.g. 123, 123, …, 123, 0, 124, 124, …
  - Must look directly at ADC output to detect
- Noise not detected or improves DNL
  - E.g. 9, 9, 9, 10, 9, 9, 9, 10, 9, 10, 10, 10, …
- Reference
  - B. Ginetti and P. Jespers, “Reliability of Code Density Test for High Resolution ADCs,” Electron. Letters, pp. 2231-2233, Nov. 1991.

Hiding Problems in the Noise

- INL looks a lot like there are 5 missing codes
- DNL "smeared out" by noise!
- Always look at both DNL/INL
- INL usually does not lie...



[Source: David Robertson, Analog Devices]