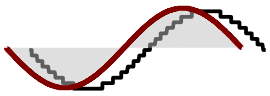
STANFORD UNIVERSITY
Department of Electrical Engineering
Prof. Boris Murmann

EE315: VLSI Data Conversion Circuits

Table of Contents

Lecture 1
Introduction
Ideal Sampling, Reconstruction



Boris Murmann
Stanford University
murmann@stanford.edu

EE315 Basics (1)

- Teaching assistants
  - Fernando Gomez (Lead TA)
  - Wei Xiong
- Administrative support
  - Ann Guerra, CIS 207
- Lectures are televised and on the web, but please come to class to keep the discussion intercative
- Web page: http://eeclass.stanford.edu/ee315
  - Check regularly, especially the "bulletin board" section
  - Only enrolled students can register for eeclass access
    - We synchronize the eeclass database with axess.stanford.edu manually, ~ once per day during first week of instruction

EE315 Basics (2)

- Course prerequisites
  - EE214 or equivalent
    - Transistor level analog circuits, including op-amp/OTA design
    - Basic MOS device physics and models
  - Prior exposure to HSpice, Matlab
  - Basic signals and systems, probability
- Please talk to me if you are not sure if you have the required background

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

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%)
  - Design of a switched capacitor stage
  - Transistor level design of sampling network
  - Noise and linearity simulations using HSpice
  - 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: HSpice, Matlab
  - You can use other tools at "own risk"
  - HSpice Basics doc and example simulation file provided in private area of web site and under /usr/class/ee315/hspice
  - From your Leland account source /usr/class/ee315/hspice/DOT.cshrc to set HSpice path
- Matlab is the preferred tool for all simulation plots
  - Include /usr/class/ee315/matlab/hspice_toolbox in your Matlab path
  - Or download Hspice toolbox at: http://www-mtl.mit.edu/research/perrottgroup/tools.html#hspice
- EE315 Technology
  - 0.18-μm CMOS
  - BSIM3v3 models provided in private area of web site and under /usr/class/ee315/hspice/lib

Course Topics

- Ideal sampling, reconstruction and quantization
- Sampling circuits
- Switched capacitor circuits
- Voltage comparators
- Nyquist-rate ADCs and DACs
- Oversampled ADCs and DACs
- Data converter performance trends and limits
- Data converter testing
- Layout considerations (time permitting)
- Filters (time permitting)

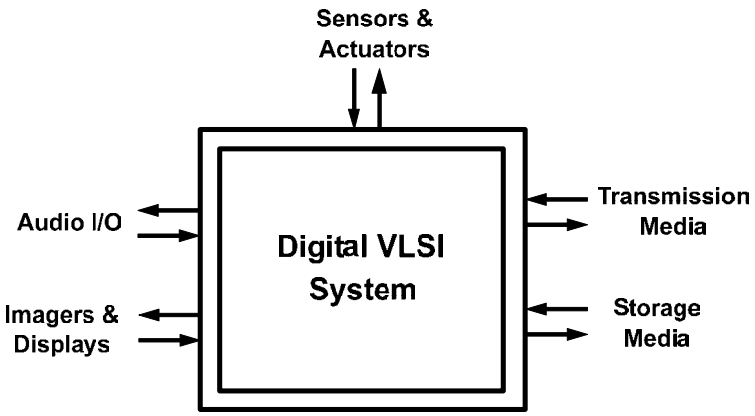
Reference Books

- 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.

Acknowledgements

- Much of the material presented in EE315 builds on course material developed previously
  - EE315 at Stanford
    - Prof. Bruce Wooley & staff
  - EE247 at UC Berkeley
    - Prof. Bernhard Boser & staff
- Notes on filters originally compiled by Susan Luschas

Motivation (1)

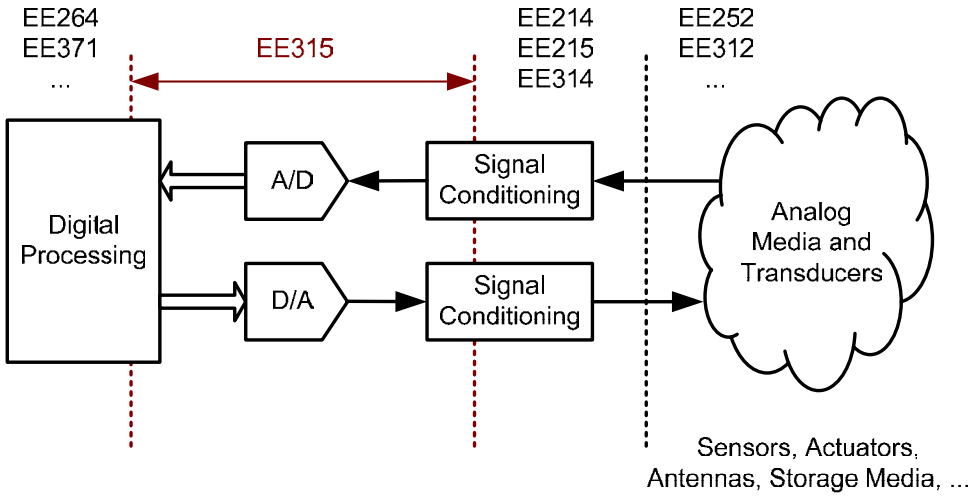


- Information is increasingly being stored, processed and communicated in digital form
- Since physical signals are analog in nature, we need A/D and D/A conversion interfaces

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

Big Picture

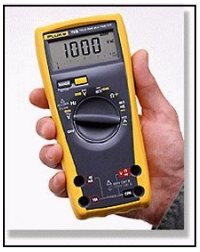


Data Converter Applications (1)

- 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



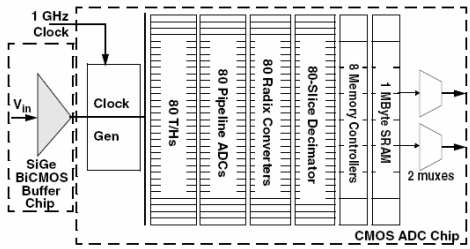
Example 1

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

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

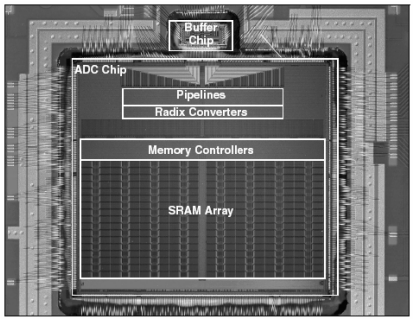
  } Audio, Tx/Rx power control, Battery charge control, display, ...
- A total of 19 data converters!

Example 2

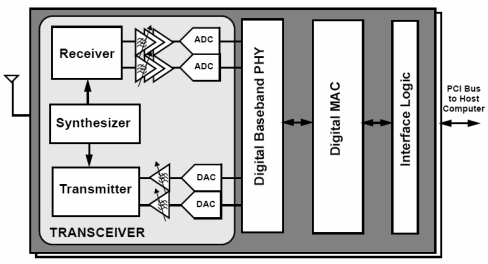


[Poulton, ISSCC 2003]

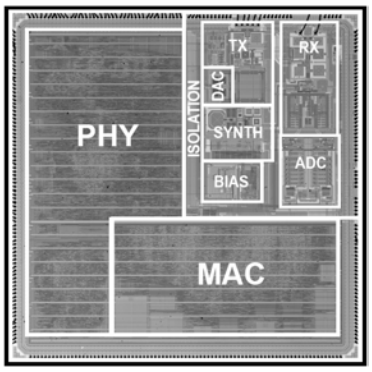
- High performance digital oscilloscopes rely on extremely high performance ADCs
- Example
  - 20 GSample/s, 8-bit ADC
  - 10 W Power dissipation



Example 3

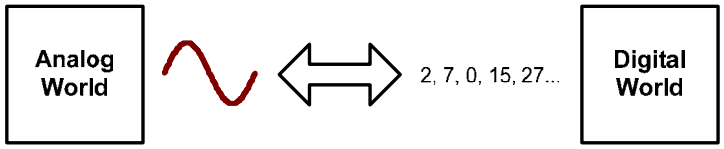


[Mehta, ISSCC2005]



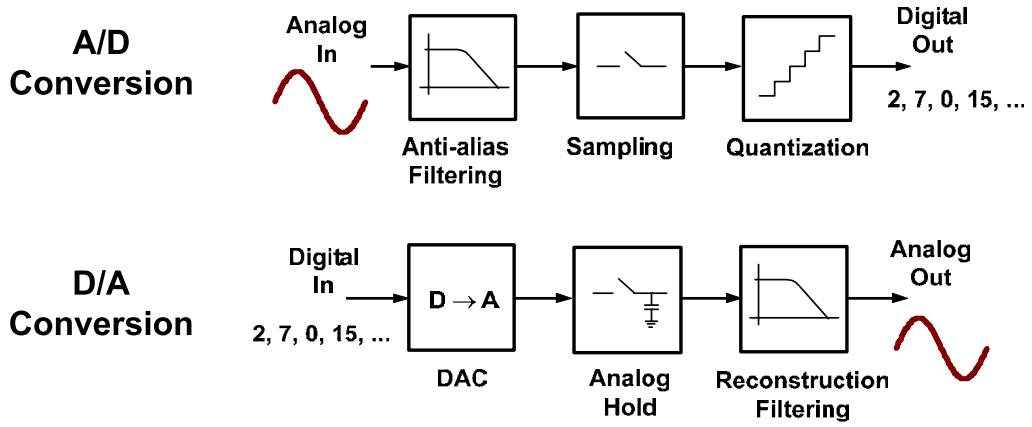
- Low-cost, single chip solutions require embedded data conversion
- Example: 802.11g Wireless LAN chip
  - 2x 11-bit DAC, 176 MSamples/s
  - 2x 9-bit ADC, 80 MSamples/s

The Data Conversion Problem



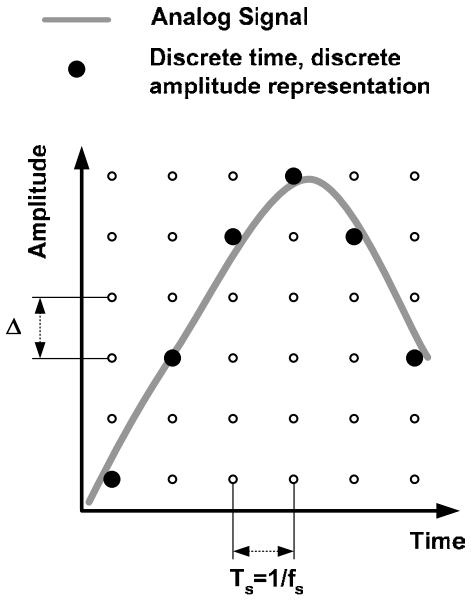
- Real world signals
  - Continuous time, continuous amplitude
- Digital abstraction
  - Discrete time, discrete amplitude
- Two problems
  - How to discretize in time and amplitude
    - A/D conversion
  - How to "undescretize" in time and amplitude
    - D/A conversion

Overview



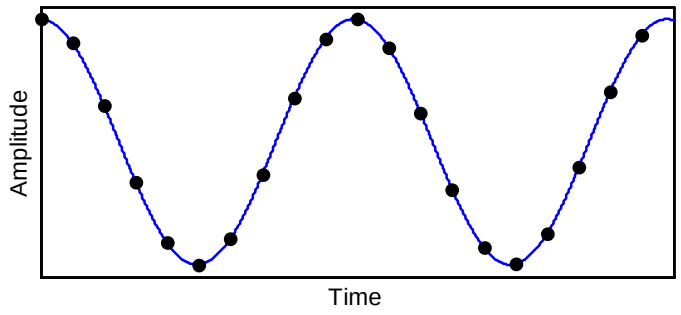
- We'll fist look at these building blocks from a functional, "black box" perspective
  - Refine later and look at implementations

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 much "noise" is added due to amplitude quantization?
  - How can we reconstruct the signal back into analog form?
  - How fast do we need to sample?
    - Must avoid "aliasing"

Aliasing Example (1)



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

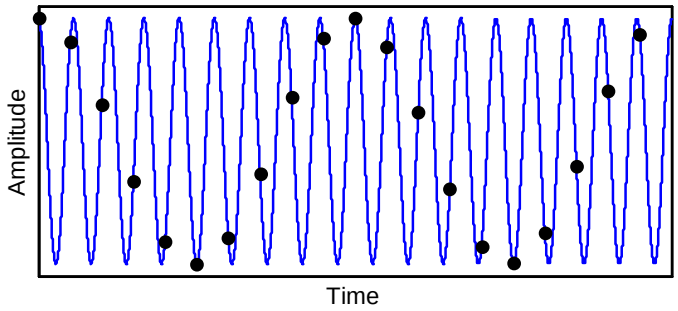
$$f_{sig} = 101kHz$$

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

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

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

Aliasing Example (2)

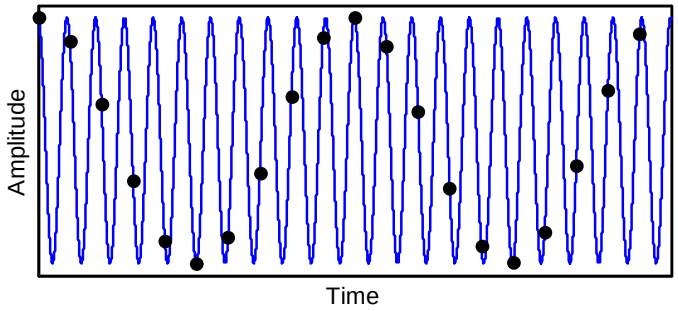


$$f_s = \frac{1}{T_s} = 1000kHz$$

$$f_{sig} = 899kHz$$

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

Aliasing Example (3)

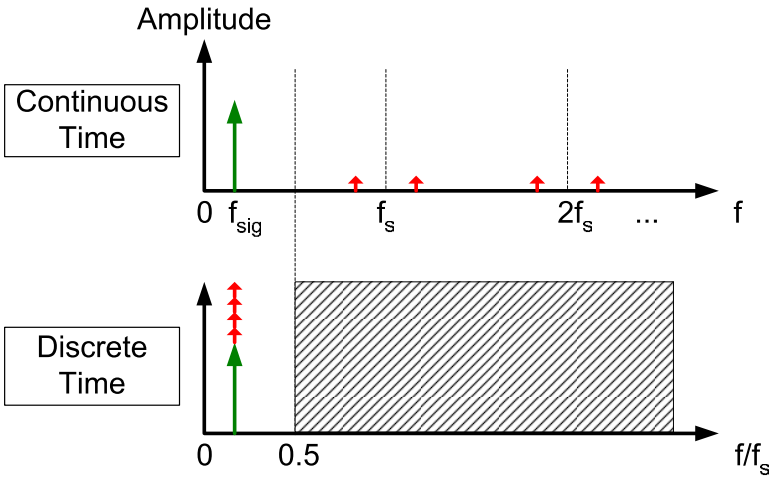


$$f_s = \frac{1}{T_s} = 1000kHz$$

$$f_{sig} = 1101kHz$$

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

Consequence



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

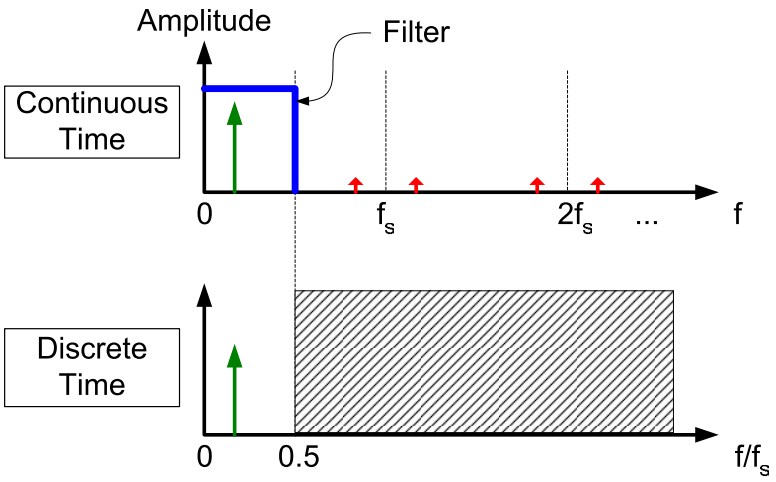
Sampling Theorem

- In order to prevent aliasing, we need

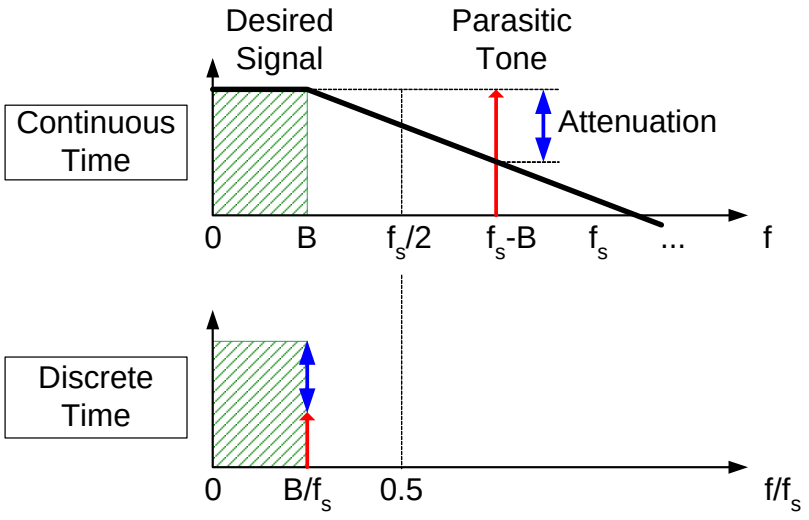
$$f_{sig,max} < \frac{f_s}{2}$$

- 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

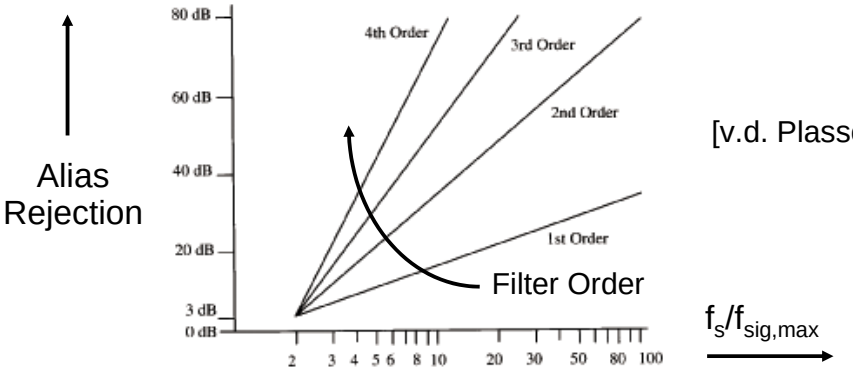


Practical Anti-Alias Filter



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

How much Oversampling?



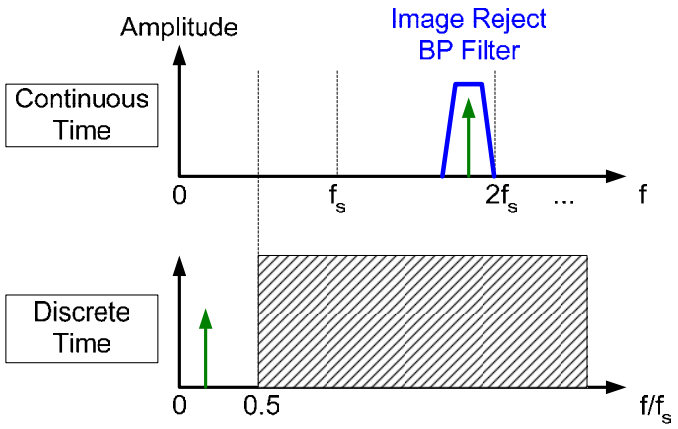
[v.d. Plassche, p.41]

- 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

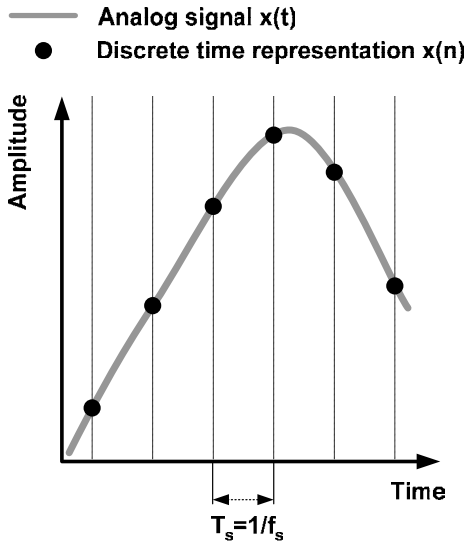
- Nyquist-rate sampling ($f_s > 2 \cdot f_{sig,max}$)
  - Nyquist data converters
  - In practice always slightly oversampled
- Oversampling ($f_s >> 2 \cdot f_{sig,max}$)
  - Oversampled data converters
  - Anti-alias filtering is often trivial
  - Oversampling also helps reduce "quantization noise"
    - 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 (!)

The Reconstruction Problem



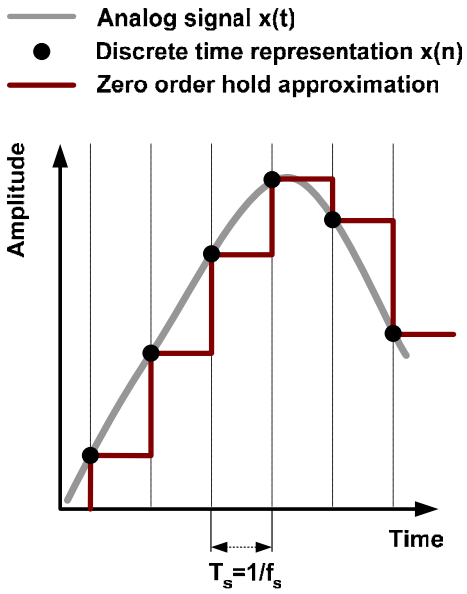
- As long as we sample fast enough, x(n) contains all information about x(t)
  - $f_s > 2 \cdot f_{sig,max}$
- How to reconstruct x(t) from x(n)?
- Ideal interpolation formula

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

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

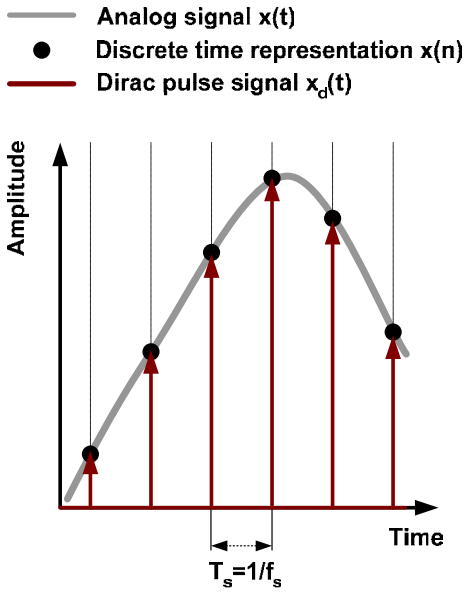
- Very hard to build an analog circuit that does this…

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
- What does this do to the signal spectrum?
- We'll analyze this in two steps
  - First look at infinitely narrow reconstruction pulses

Dirac Pulses



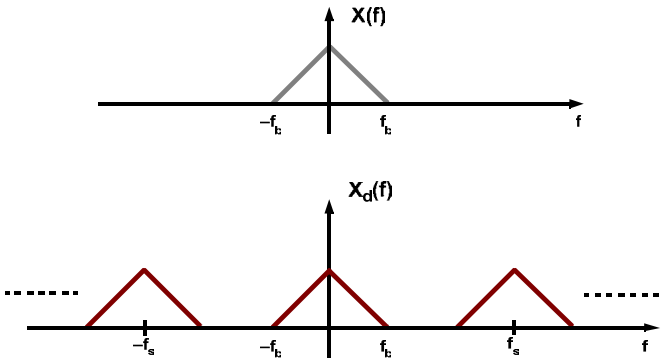
- $x_d(t)$ is zero between pulses
  - Note that x(n) is undefined at these times

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

- Multiplication in time means convolution in frequency
  - Resulting spectrum

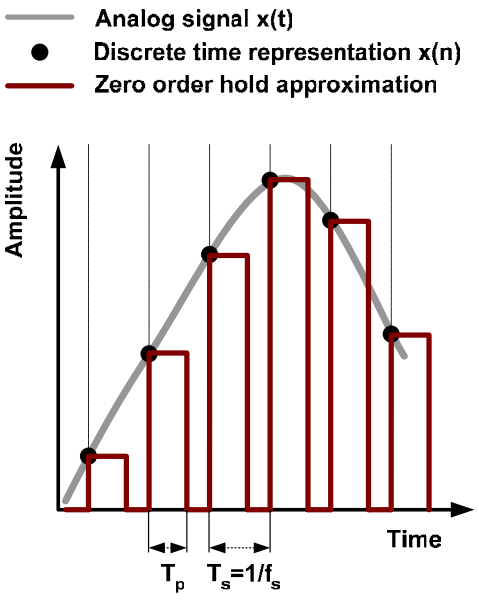
$$X_d(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X\left(f - \frac{n}{T_s}\right)$$

Spectrum



- Spectrum contains replicas of X(f) at integer multiples of the sampling frequency

Finite Hold Pulse



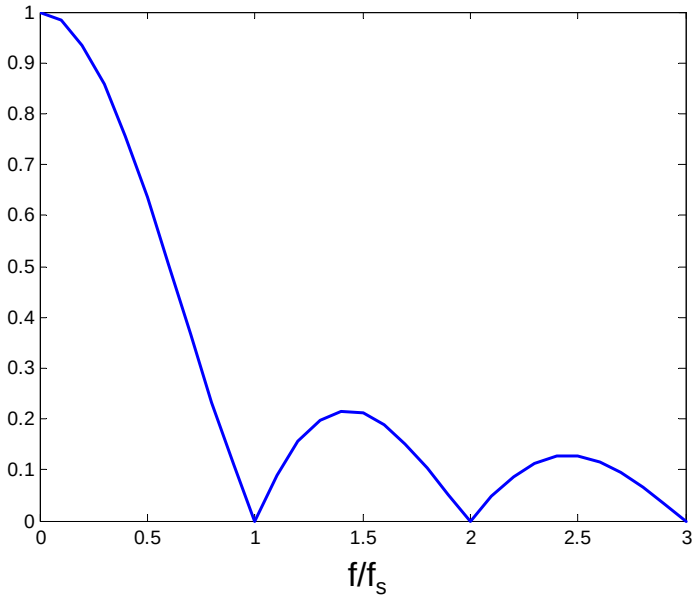
- Consider the general case with a rectangular pulse $0 < T_p \leq T_s$
- The time domain signal on the left follows from convolving the Dirac sequence with a rectangular unit pulse
- Spectrum follows from multiplication with Fourier transform of the pulse

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

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

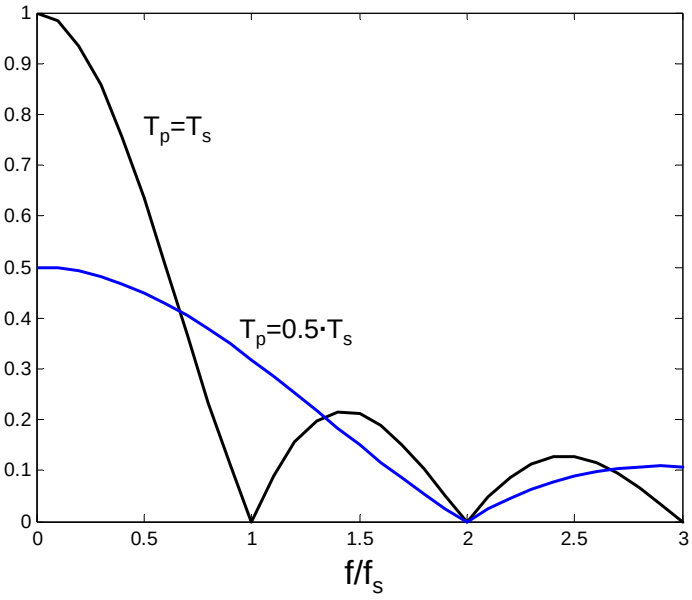
Envelope with Hold Pulse $T_p=T_s$

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

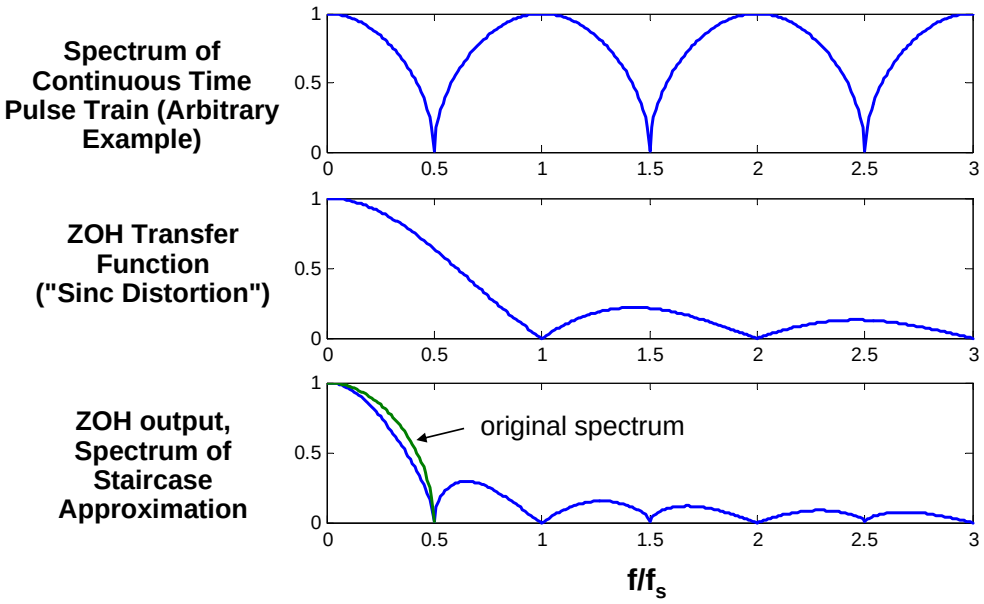


Envelope with Hold Pulse $T_p=0.5\cdot T_s$

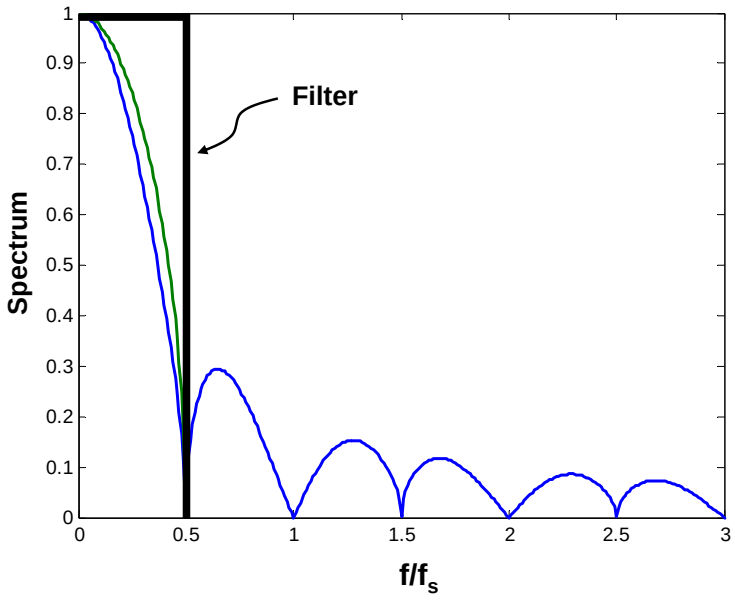
$$\frac{T_p}{T_s}\frac{\sin(\pi f T_p)}{\pi f T_p}$$



Example



Reconstruction Filter

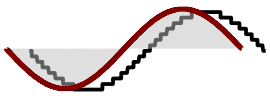


- Also called smoothing filter
- Same situation as with anti-alias filter
  - A brick wall filter would be nice
  - Oversampling helps reduce filter order

Summary

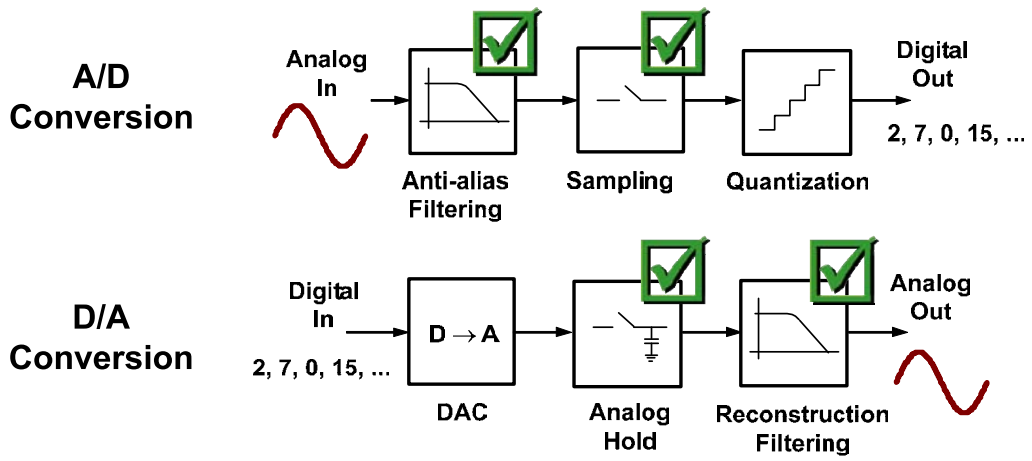
- Must obey sampling theorem $f_s > 2 \cdot f_{sig,max,}$
  - Usually dictates anti-aliasing filter
- If sampling theorem is met, continuous time signal can be recovered from discrete time sequence without loss of information
- A zero order hold in conjunction with a reconstruction filter is the most 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

Lecture 2
Quantization
Static Performance Metrics



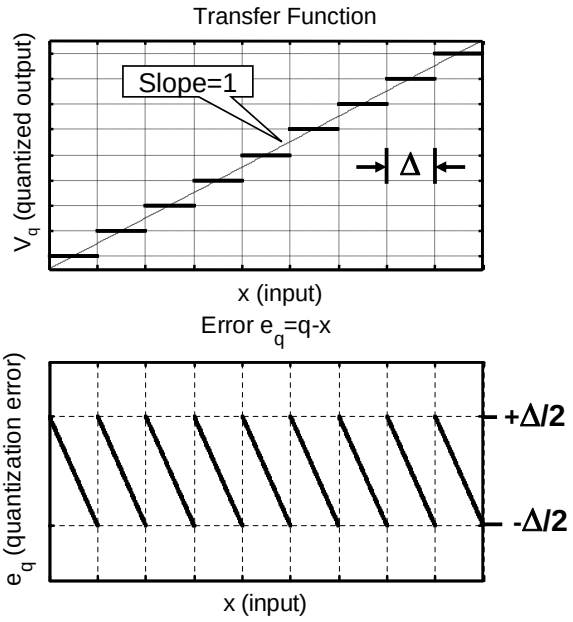
Boris Murmann
Stanford University
murmann@stanford.edu

Recap



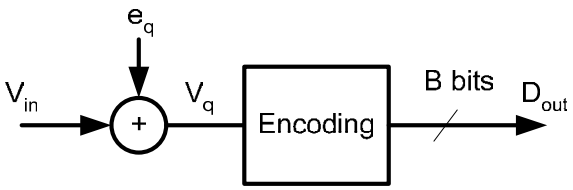
- Next, look at
  - Transfer functions of quantizer and DAC
  - Impact of quantization error

Quantization of an Analog Signal



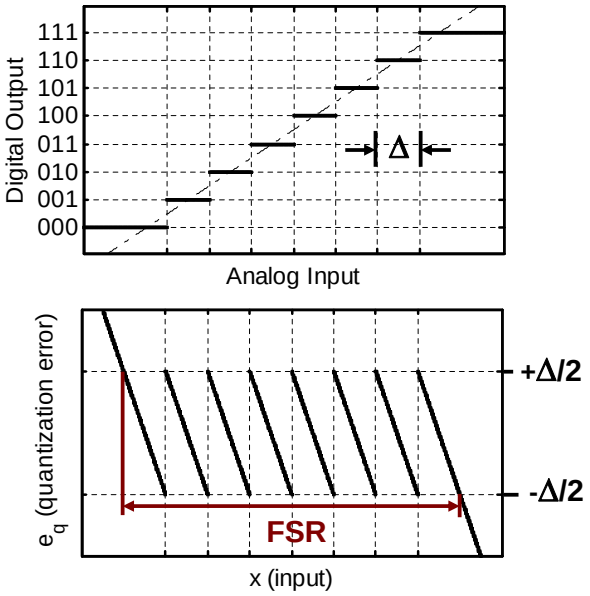
- Quantization step Δ
- Quantization error has sawtooth shape
  - Bounded by –Δ/2, +Δ/2
- Ideally
  - Infinite input range and infinite number of quantization levels
- In practice
  - Finite input range and finite number of quantization levels
  - Output is a digital word (not an analog voltage)

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

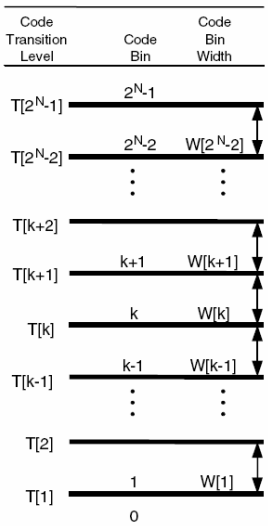
Encoding Example for a B-Bit Quantizer



- 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

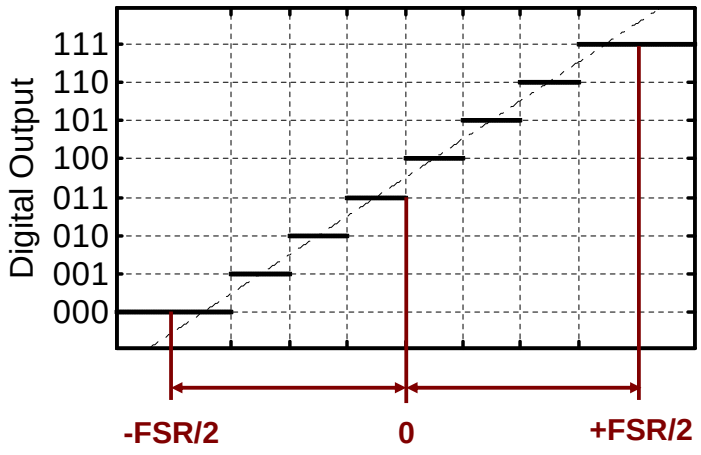


[IEEE Standard 1241-2000]

Implementation Specific Technicalities

- On slide 5, 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

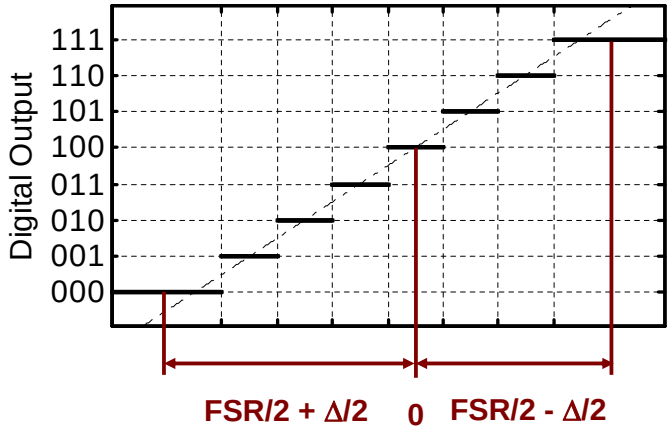
Bipolar Mid-Rise Quantizer



- Nothing new here…

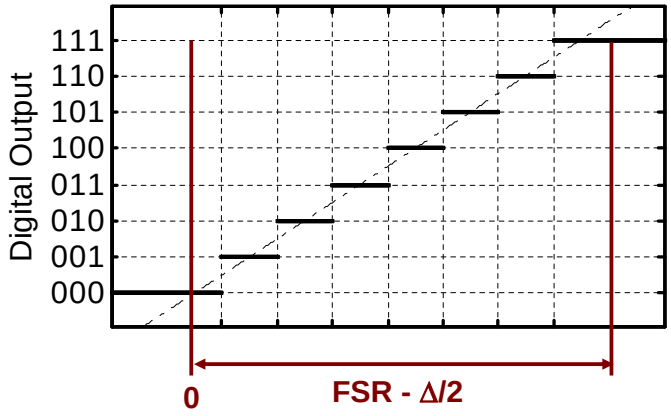
Bipolar Mid-Tread Quantizer

- 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

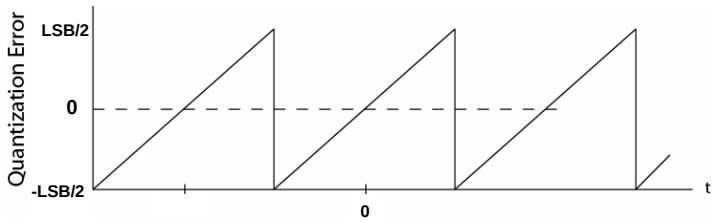


Effect of Quantization Error on Signal

- 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

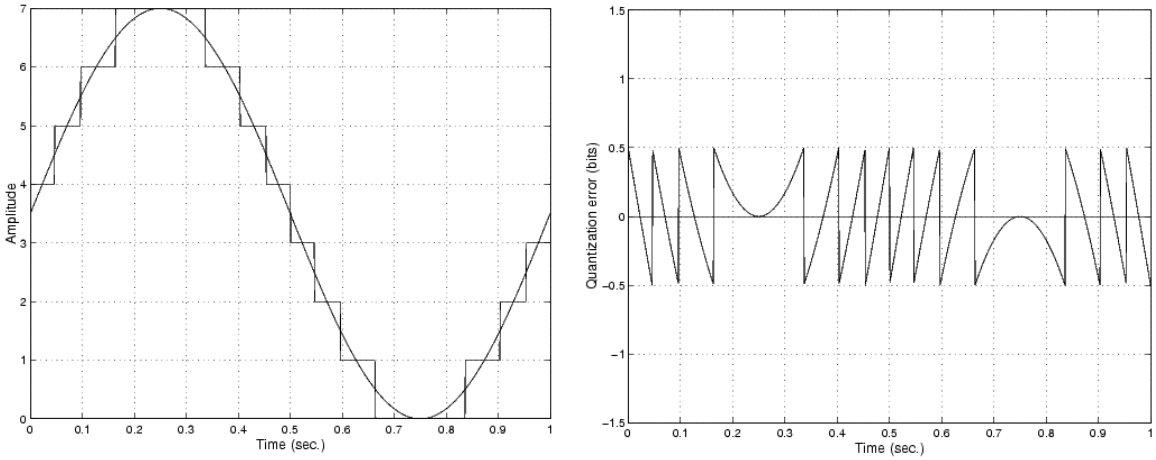
- Applying a ramp signal (periodic sawtooth) at the input of the quantizer gives the following time domain waveform for e_q



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

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

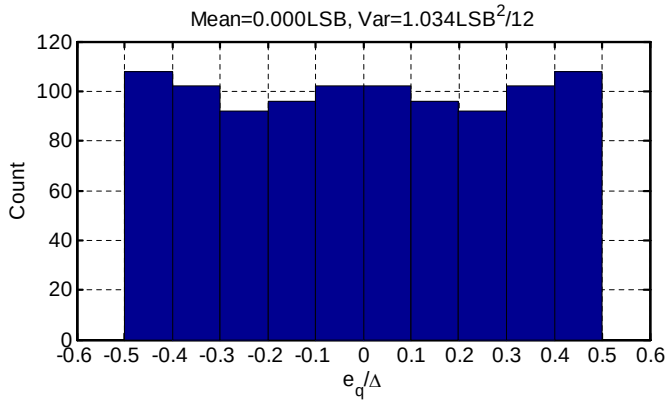
Sine Wave Input



- Integration is not straightforward…

Quantization Error Histogram

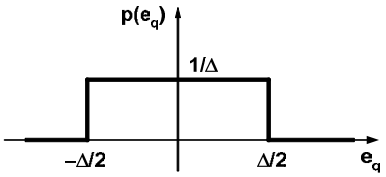
- Sinusoidal input signal with f_{sig}=101Hz, sampled at f_s=1000Hz
- 8-bit quantizer



- 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

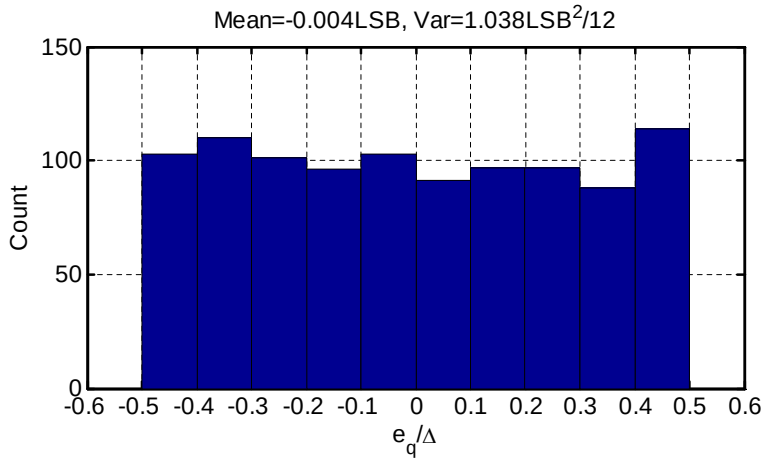


Mean $\overline{e_q} = \int_{-\Delta/2}^{+\Delta/2} \frac{e_q}{\Delta} de_q = 0$

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

Reality Check (1)

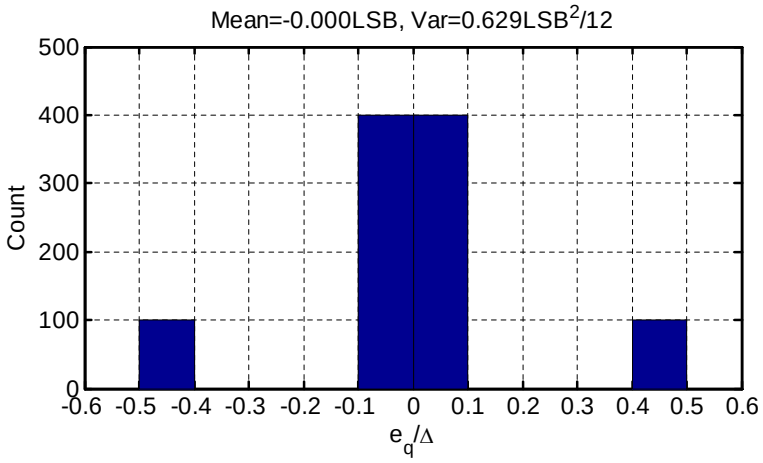
- Input sequence consists of 1000 samples drawn from Gaussian distribution, 4σ=FSR



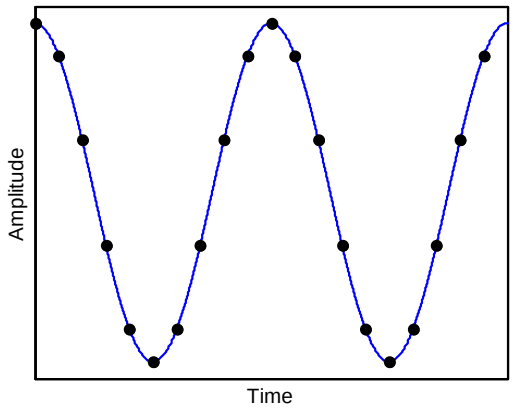
- Error power close to that of uniform approximation

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

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

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

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

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

Signal-to-Quantization-Noise Ratio

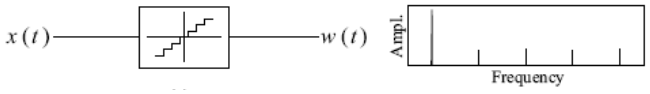
- Assuming uniform distribution of e_q and a full-scale sinusoidal input, we have

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

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)



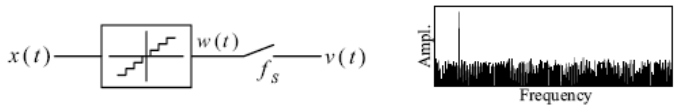
+ many more harmonics

[Y. Tsividis, ICASSP 2004]

- Quantization results in an "infinite" number of harmonics

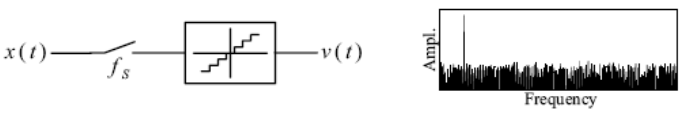
Quantization Noise Spectrum (2)

- 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"



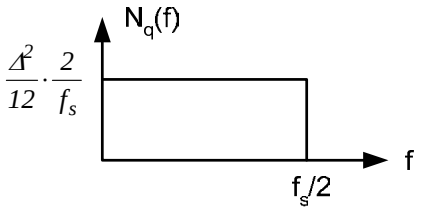
[Y. Tsividis, ICASSP 2004]

- Interchanging sampling and quantization won't change this situation



Quantization Noise Spectrum (3)

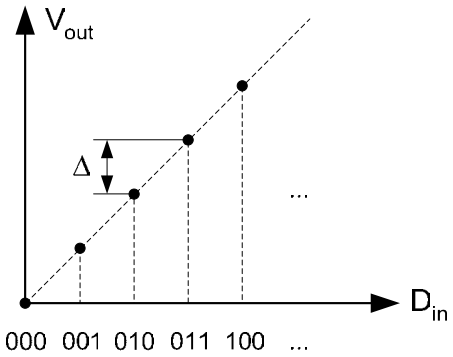
- Can show that the quantization noise power is indeed distributed (approximately) uniformly in frequency
  - Again, this is provided that the quantization error is "sufficiently random"



- 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.

Ideal DAC

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



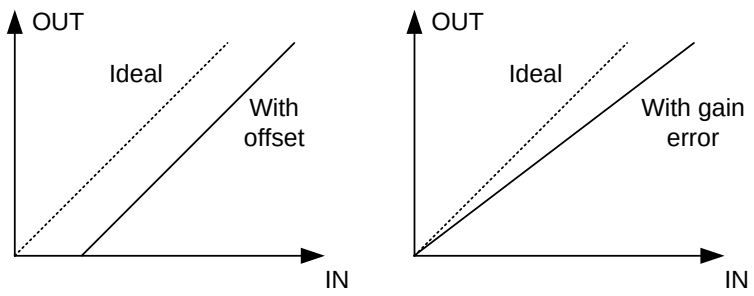
Static Nonidealities

- 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

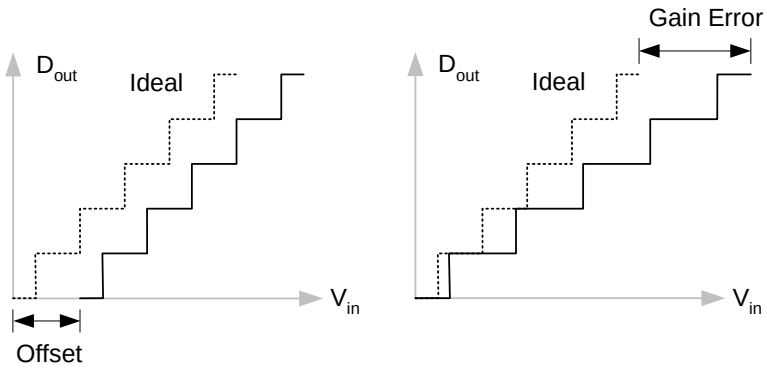
Offset and Gain Error

- 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):



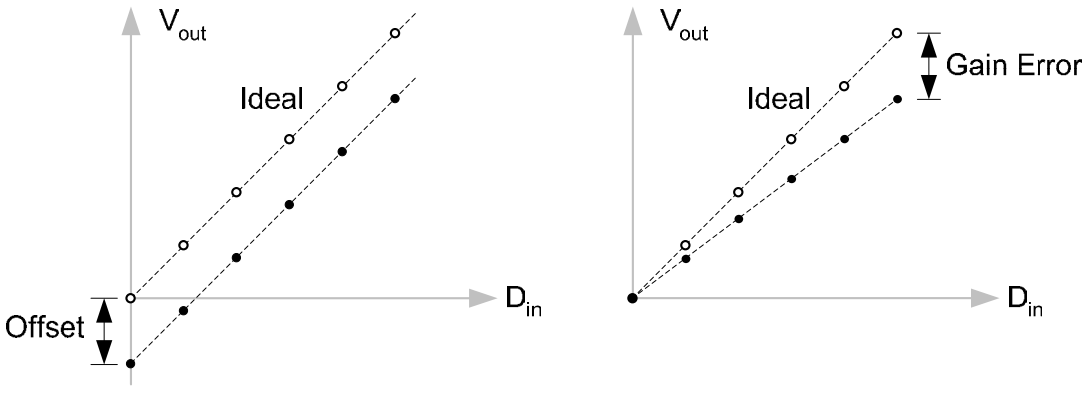
ADC Offset and Gain Error

- 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



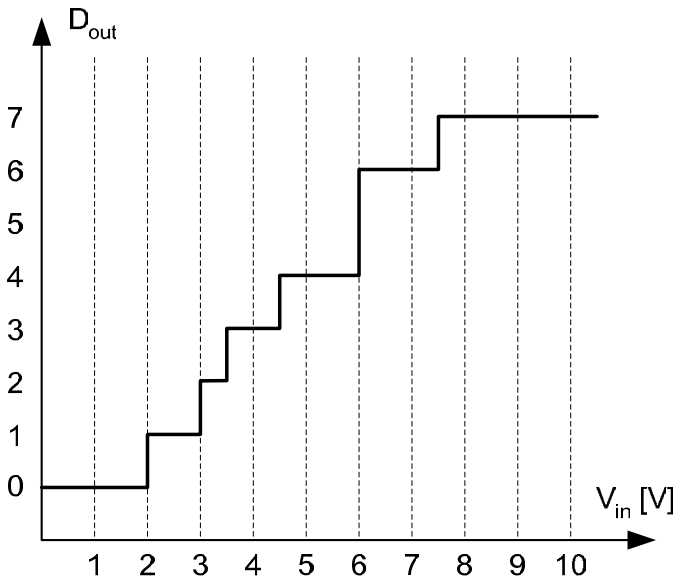
Comments on Offset and Gain Errors

- 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

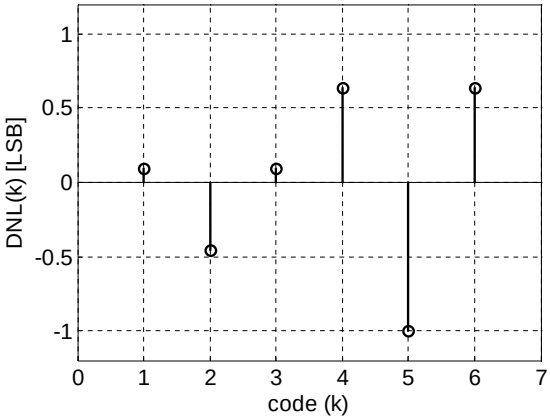
$$W_{avg} = \frac{7.5V - 2V}{6} = 0.9167V$$

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

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

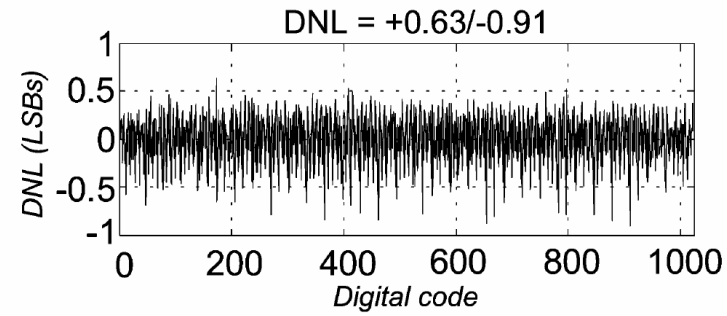
Result

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

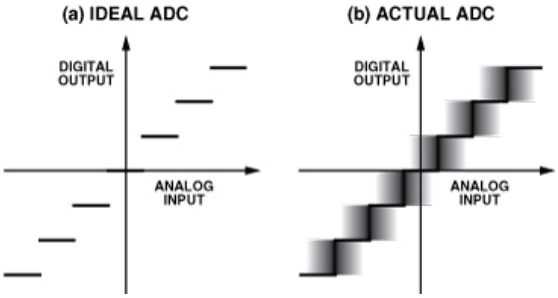
A Typical ADC DNL Plot



[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

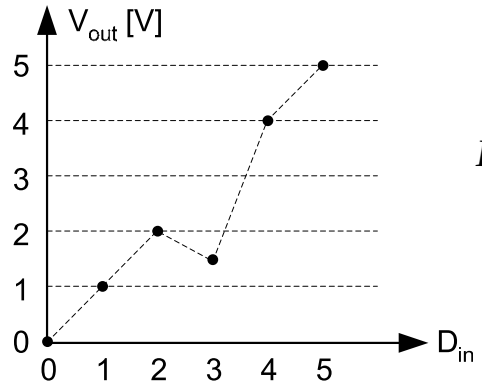
DAC DNL

- 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

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

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

Non-Monotonic DAC

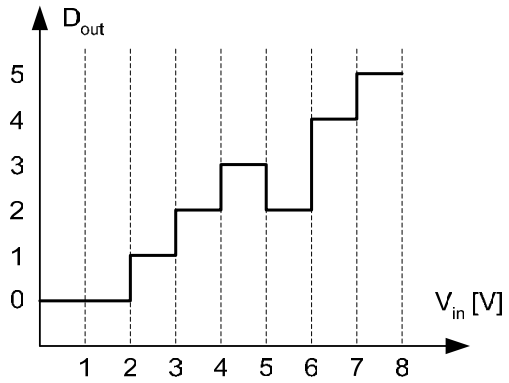


$$DNL(3) = \frac{Step(3) - Step_{avg}}{Step_{avg}}$$

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

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

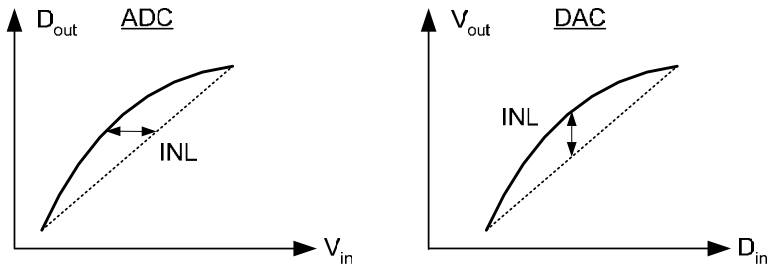
Non-Monotonic ADC



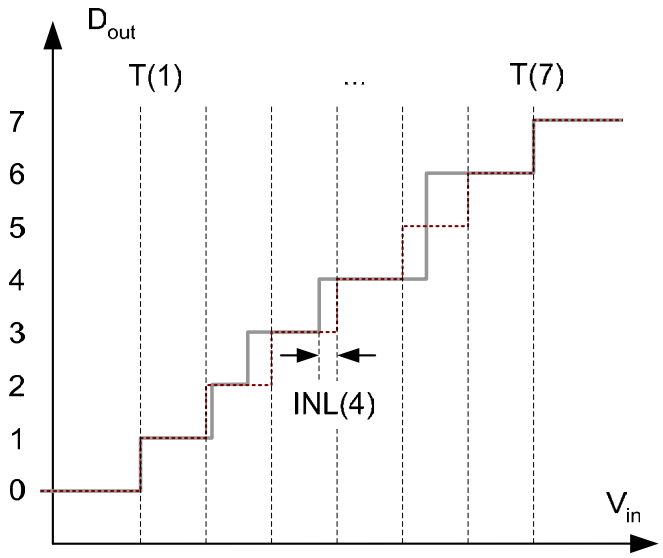
- Code 2 has two transition levels $\Rightarrow$ W(2) is ill defined
  - DNL is ill-defined!
- Not a very big issue, because a non-monotonic ADC is usually not what we'll design for in practice…

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

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

- Obvious that INL(1) = 0 and INL(7) = 0
- INL(0) is undefined

ADC INL Example (2)

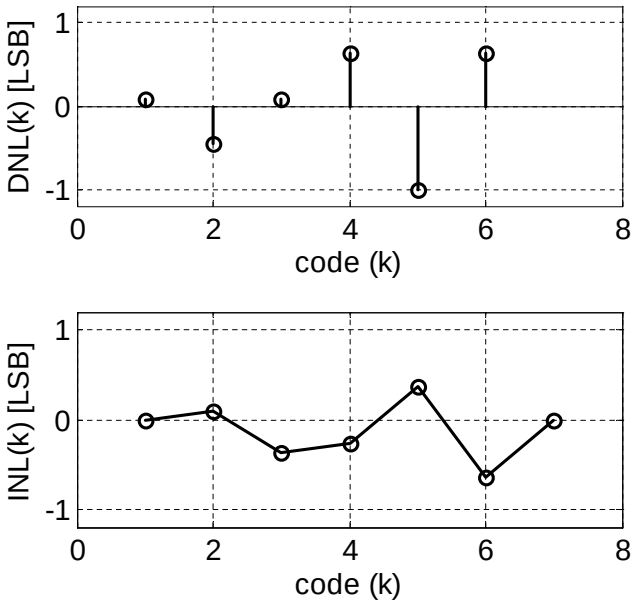
- Can show that

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

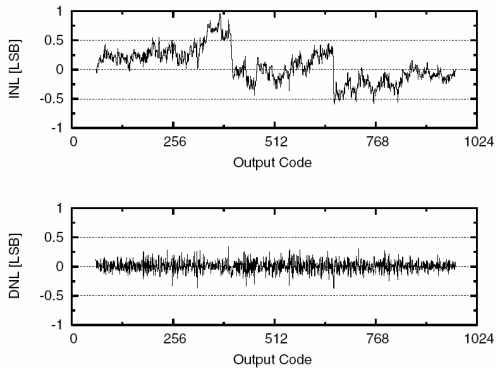
- Means that once we computed DNL, we can easily find INL using a cumulative sum operation on the DNL vector
- Using DNL values from last lecture, we find

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]

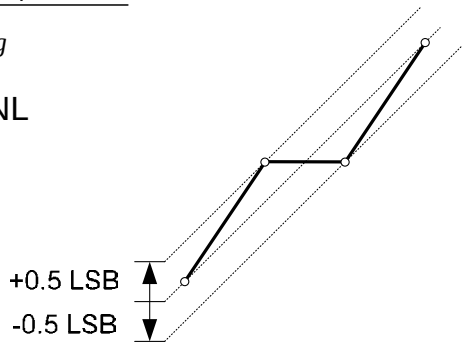
- DNL/INL signature often reveals architectural details
  - E.g. major transitions
  - We'll see more examples in the context of DACs
- Since INL is a cumulative measure, it turns out to be less sensitive than DNL to thermal noise "smearing"

DAC INL

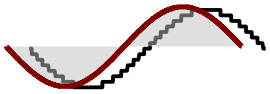
- Same idea applies
  - Find ideal output values that lie on a straight line between endpoints
  - Calculate INL for each code k using

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

- 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



Lecture 3
Spectral Performance Metrics



Boris Murmann
Stanford University
murmann@stanford.edu

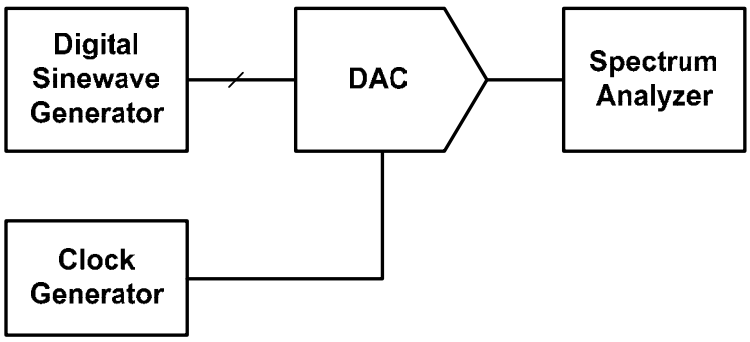
Dynamic Performance Metrics

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

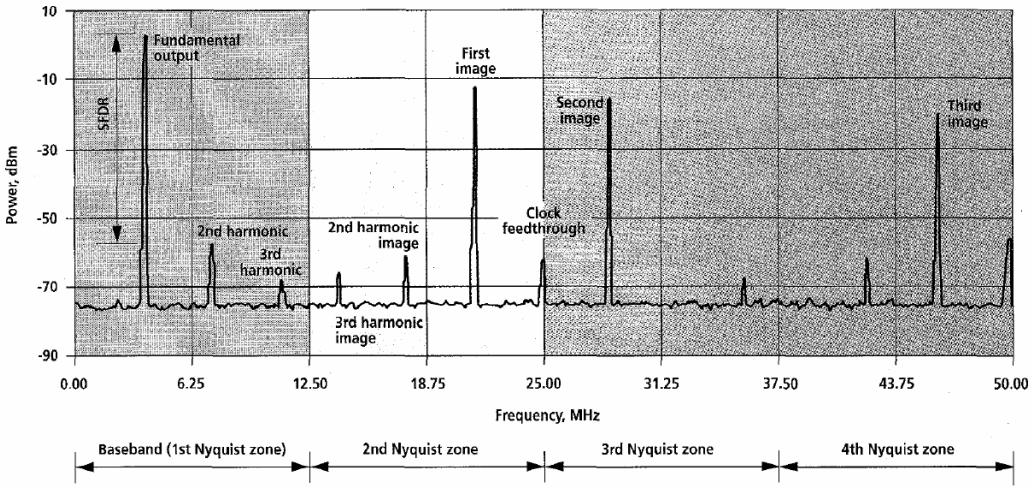
Alphabet Soup of Spectral Metrics

- 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

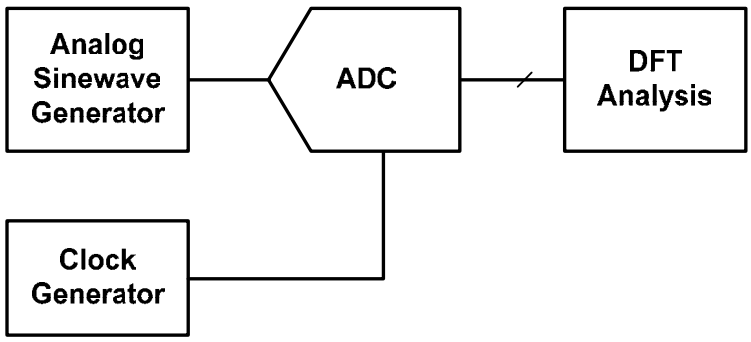


Typical DAC Output Spectrum



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

ADC Tone Test/Simulation



Discrete Fourier Transform Basics

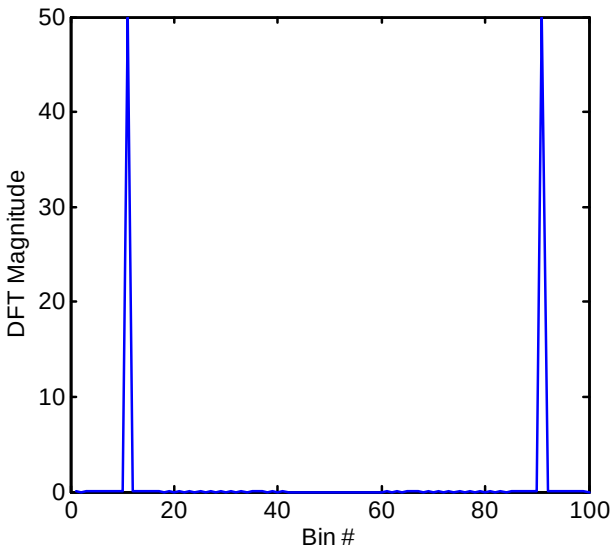
- DFT takes a block of N time domain samples (spaced $T_s=1/f_s$) and yields a set of N frequency bins

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

- 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

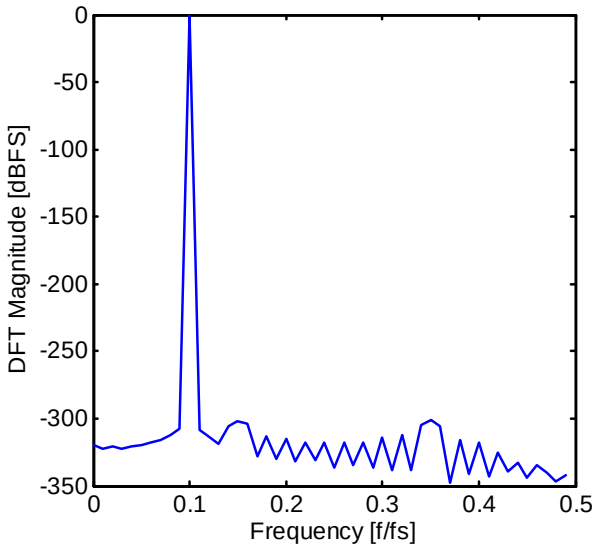
Matlab Example

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

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



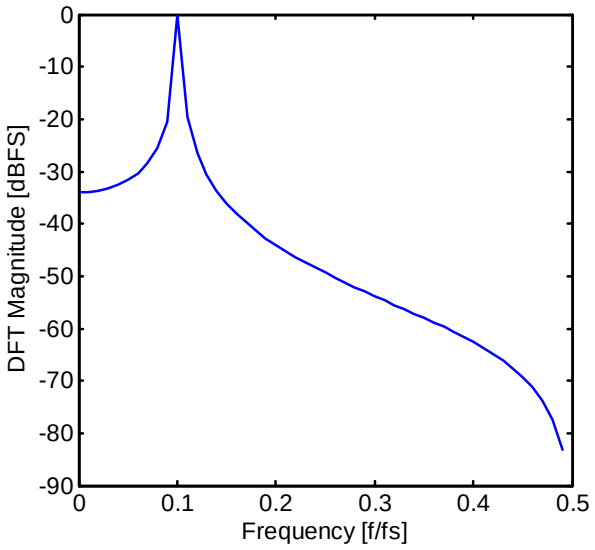
Normalized Plot with Frequency Axis

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

x = A*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
s = 20*log10(s/A/N*2);
%frequency vector
f = [0:N/2-1]/N;

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



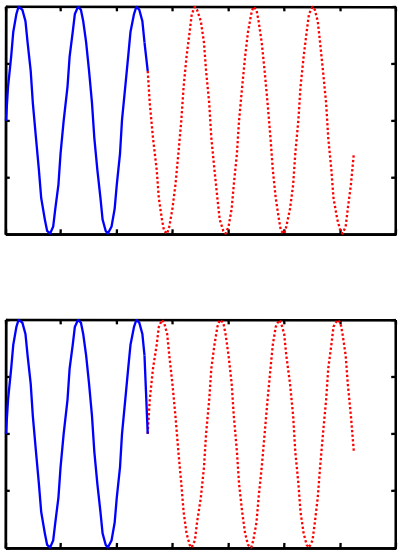
Another Example

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



Spectral Leakage

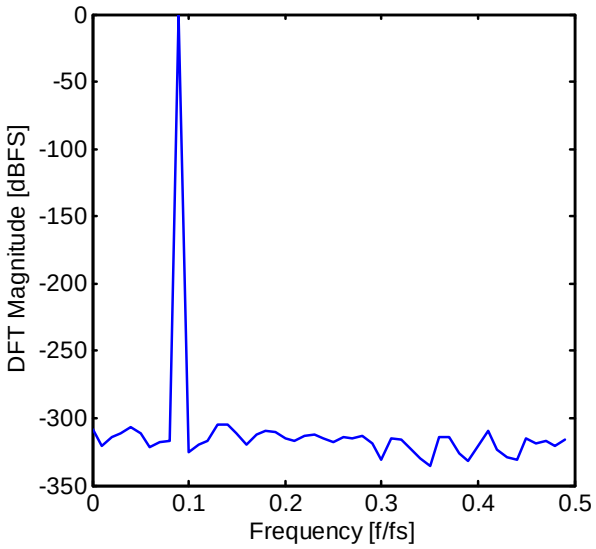
- DFT implicitly assumes that data repeats every N samples
- A sequence that contains a non-integer number of sine wave cycles has discontinuities in its periodic repetition
  - Discontinuity looks like a high frequency signal component
  - Power spreads across spectrum
- Two ways to deal with this
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

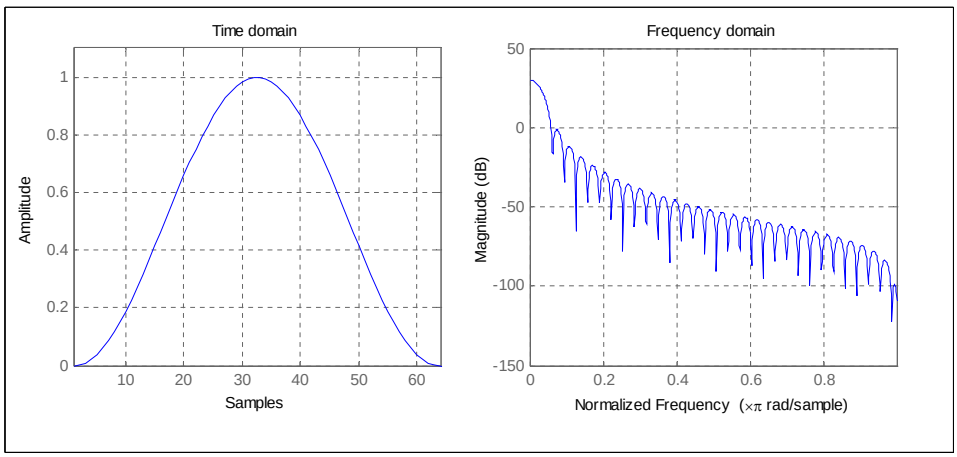


Windowing

- 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

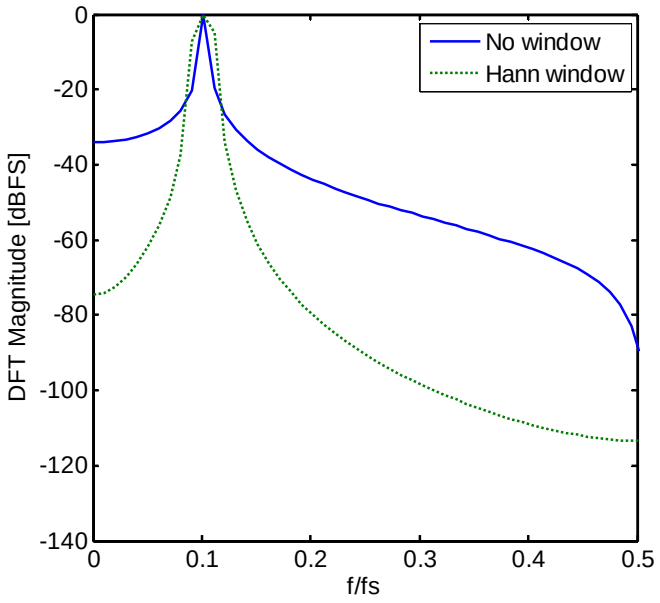
Hann Window

```
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

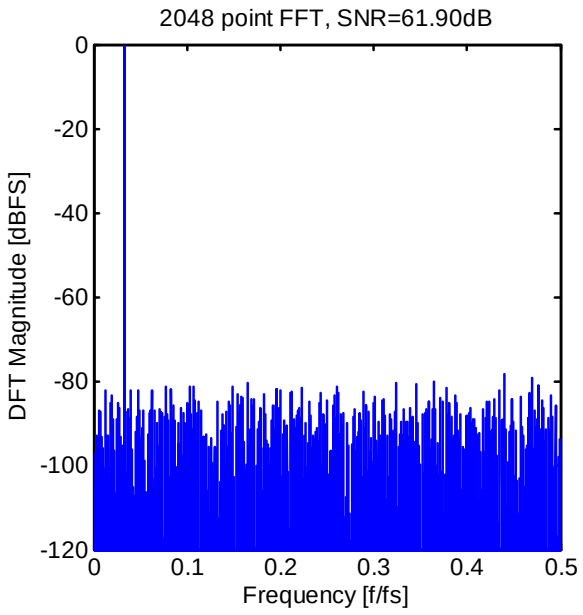
Example

- 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 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) );
```

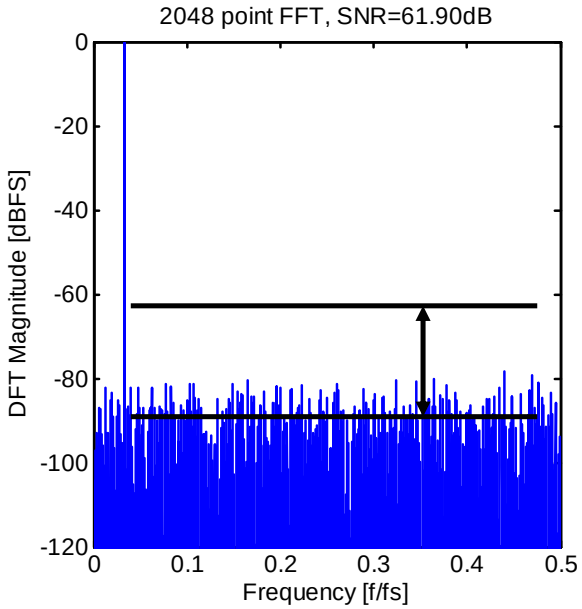

Spectrum with Quantization Noise

- Spectrum looks fairly uniform
- Signal-to-quantization noise ratio is given by power in signal bin, divided by sum of all noise bins
- Expecting SQNR= 10·6.02dB +1.76dB = 61.96dB
- Noise floor of spectrum is around -80dBFS
  - Why not -62dB?



Why is Noise Floor below -62dBFS ?

- Total noise is spread over N/2 bins
- Assuming a uniform noise spectrum, this means that each bins contains 2/N times total noise power
- Noise floor is $10\log_{10}(N/2)$dB below SQNR value
  - $10\log_{10}(2048/2)=30$dB
- Peaks above predicted noise floor are due to non-uniform distribution of quantization noise

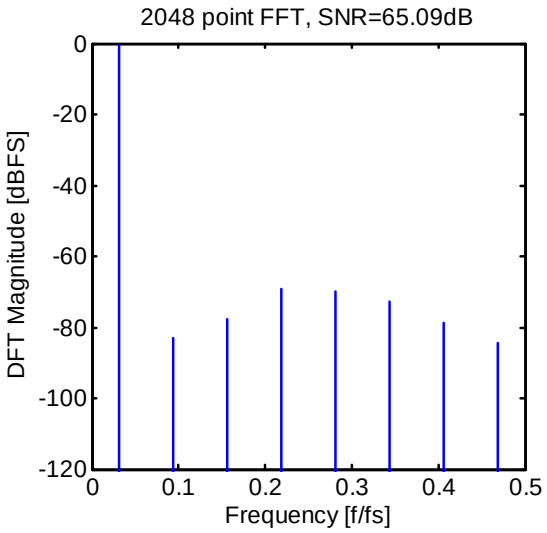


DFT Plot Annotation

- 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 determinisitc and periodic
- For more random and "white" quantizion noise, it is best to make N and cycles mutually prime
  - GCD(N,cycles)=1

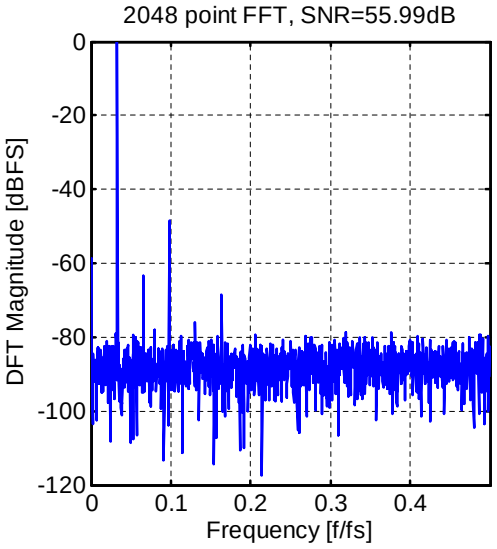


Typical ADC Output Spectrum

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

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

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



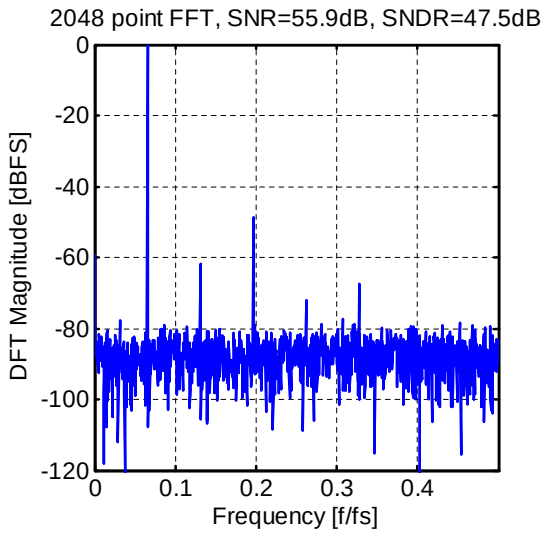
SNDR and ENOB

- Definition

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

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

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



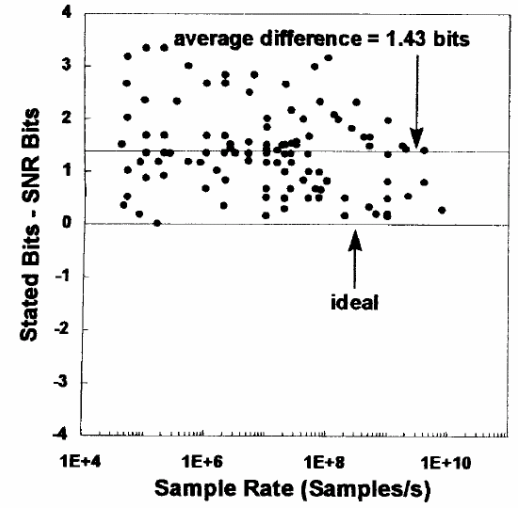
Effective Number of Bits

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

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

- 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"

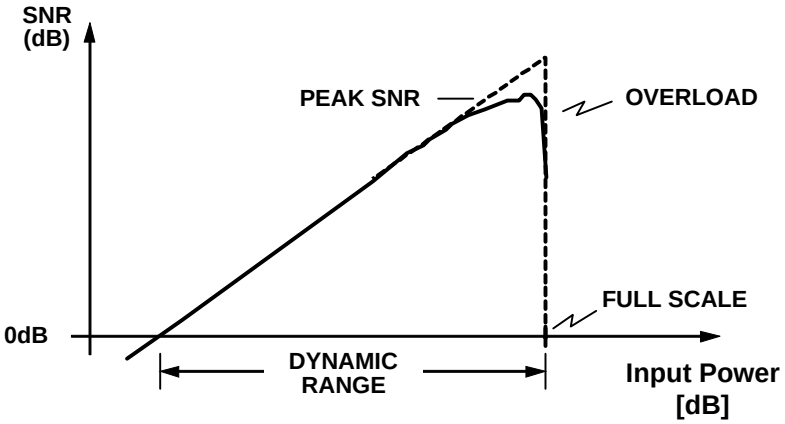
ENOB Survey



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

Dynamic Range

$$DR = \frac{Max.\,Signal\ Power}{Min.\,Signal\ Power(SNR = 0dB)} = \frac{Max.\,Signal\ Power}{Noise\ Power}$$



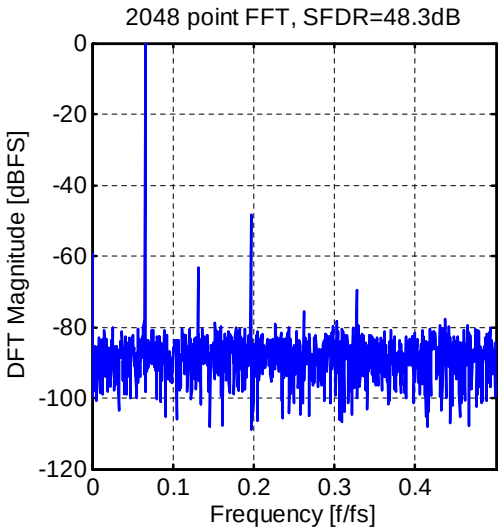
- Peak SNR ≤ DR

SFDR

- Definition of "Spurious Free Dynamic Range"

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

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

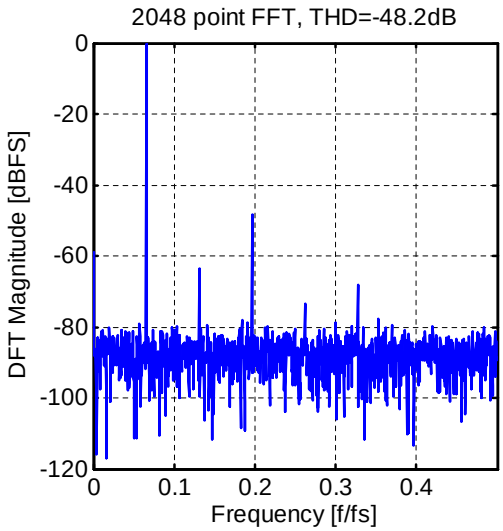


THD

- Definition

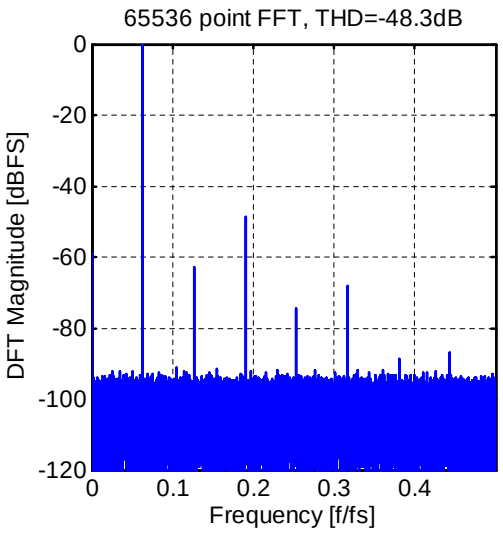
$$THD = \frac{Total\ Distortion\ Power}{Signal\ Power}$$

- By convention, total distortion power consists of 2nd through 7th harmonic
- Actually, 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

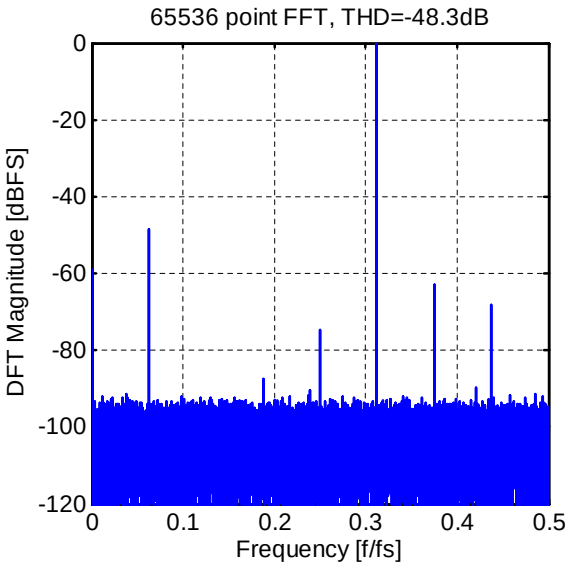
$f_1 = f_x = 0.3125\ f_s$

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

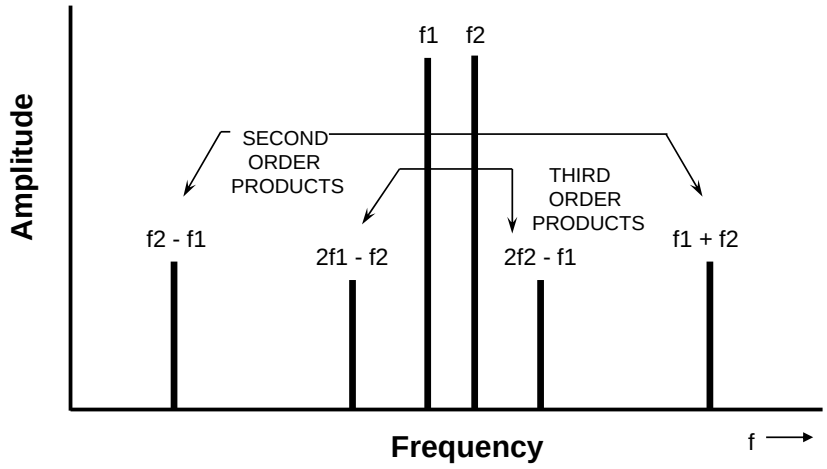
$f_3 = 3\ f_1 = 0.9375\ f_s \rightarrow 0.0625\ f_s$

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

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

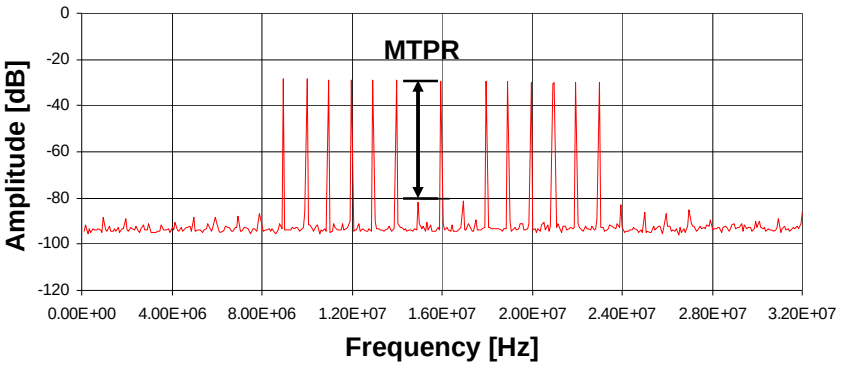


Intermodulation Distortion



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

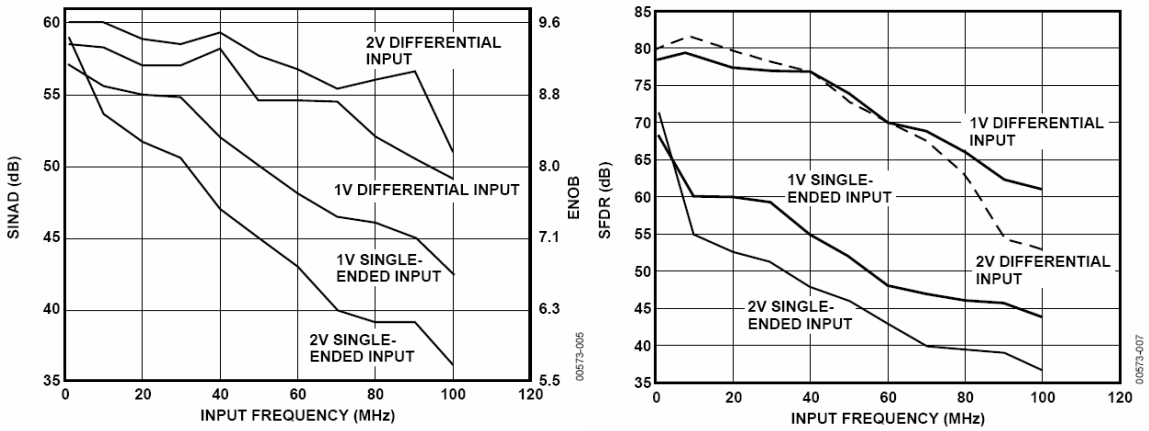
MTPR



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

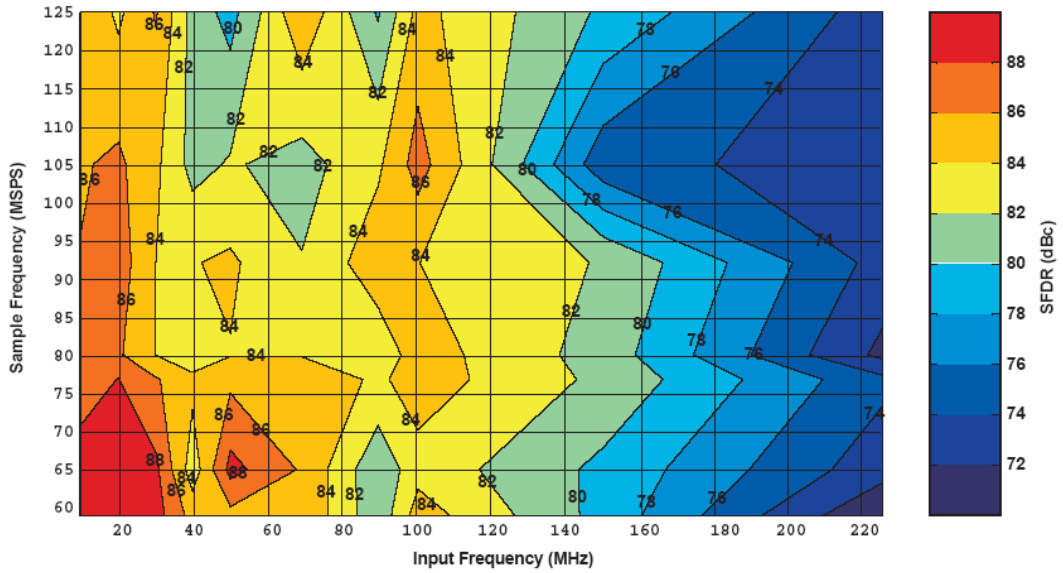
Frequency Dependence (1)

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



[Analog Devices, AD9203 Datasheet]

Frequency Dependence (2)



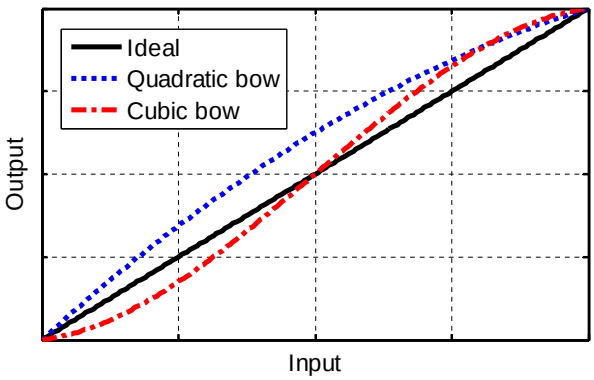
[Texas Instruments, ADS5541 Datasheet]

ERBW

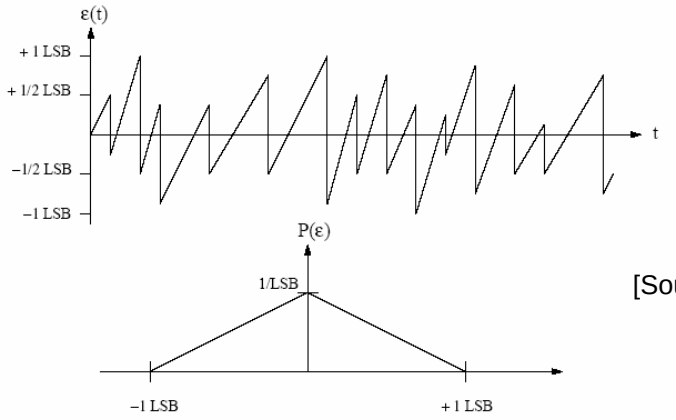
- Defined as the input frequency at which the SNDR of a converter has dropped by 3dB
  - Equivalent to a 0.5-bit loss in ENOB
- ERBW > $f_s/2$ is not uncommon, especially in converters designed for sub-sampling applications

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 ≅ $20\log(2^B/\text{INL})$
  - E.g. 1 LSB INL, 10 bits → SFDR ≅ 60dB
  - See HW2 for a more elaborate analysis



SNR Degradation due to DNL (1)



[Source: Ion Opris]

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

SNR Degradation due to DNL (2)

- Integrate triangular pdf

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

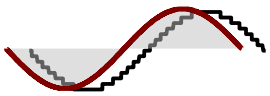
- Compare to ideal quantizer

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

3dB

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

Lecture 4
Nyquist Rate DACs

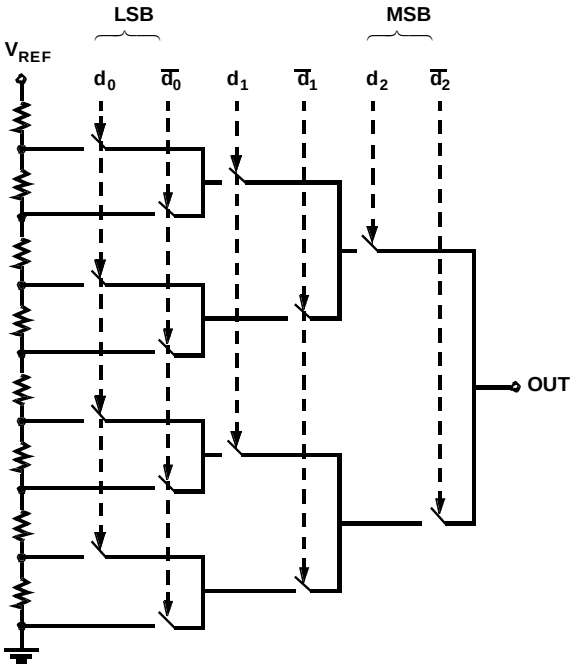


Boris Murmann
Stanford University
murmann@stanford.edu

Overview

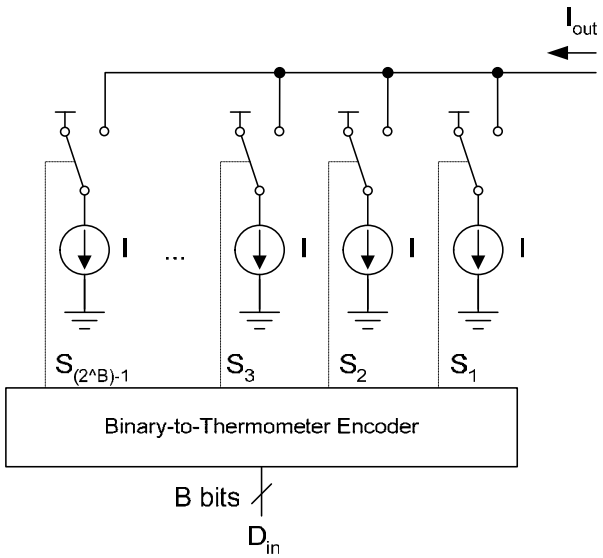
- D/A conversion is typically accomplished through the division or multiplication of a reference voltage, current or charge
- Architectures
  - Thermometer
  - Binary weighted
  - Segmented
- Static performance
  - Limited by component matching
- Dynamic performance
  - Limited e.g. by timing errors,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

Resistor String DAC



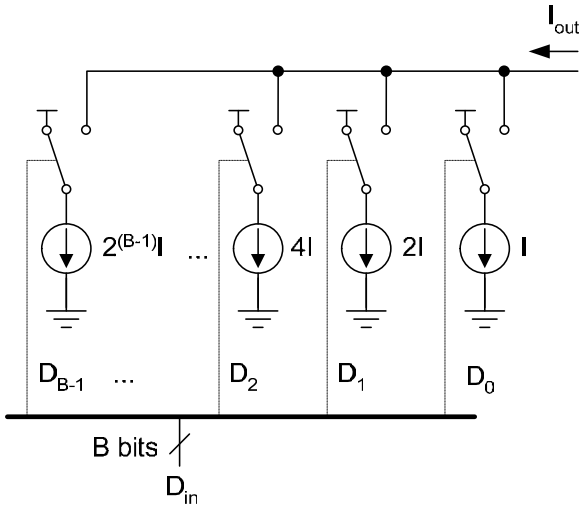
- Simple, inherently monotonic
- Small area up to ~8 bits
- See e.g. Pelgrom, JSSC 12/1990
- Unsuitable for high-resolution, high-speed designs

Thermometer DAC Using Switched Currents



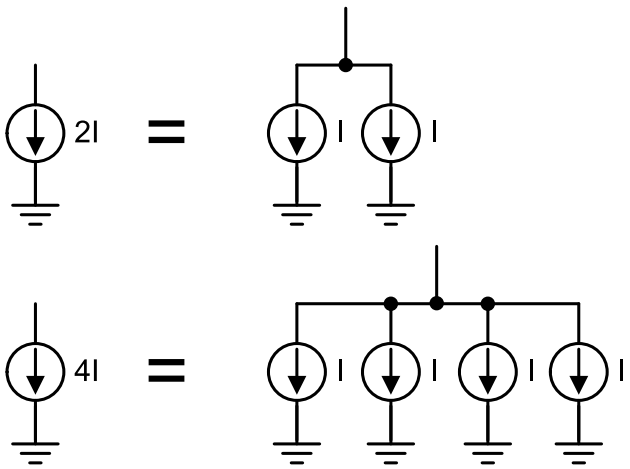
- Inherently monotonic
- Need large encoder with 2^B-1 outputs
  - Impractical for large B (high resolution)

Binary Weighted DAC

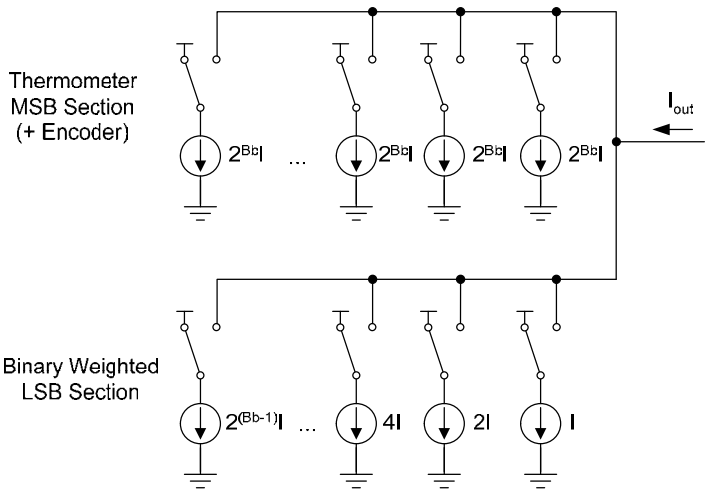


- No encoder needed
- Monotonicity is not guaranteed
- Consider transition 100000…. to 011111….
  - 2^{B-1} source must match sum of others to within 1 LSB to make transition monotonic

Implementation of Weighted Elements



Segmented DAC



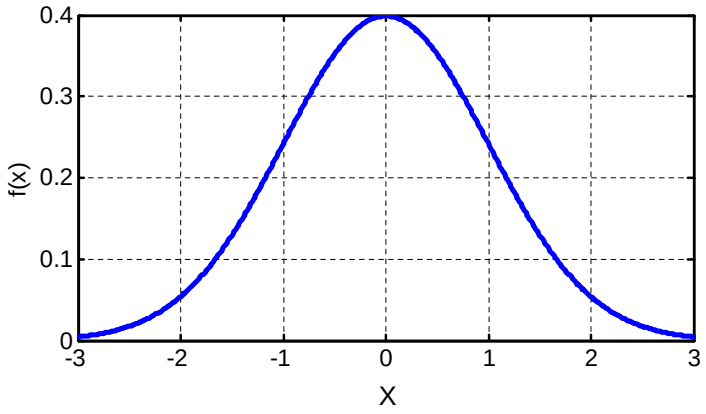
- Binary weighted section with B_b bits
- Thermometer section with $B_t = B-B_b$ bits
- Typically $B_t \sim 4…8$
- Reasonably small encoder
- Easier to achieve monotonicity

Static Errors (DNL and INL)

- Mostly 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
- Random Errors
  - Lithography
  - Often Gaussian distribution (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.

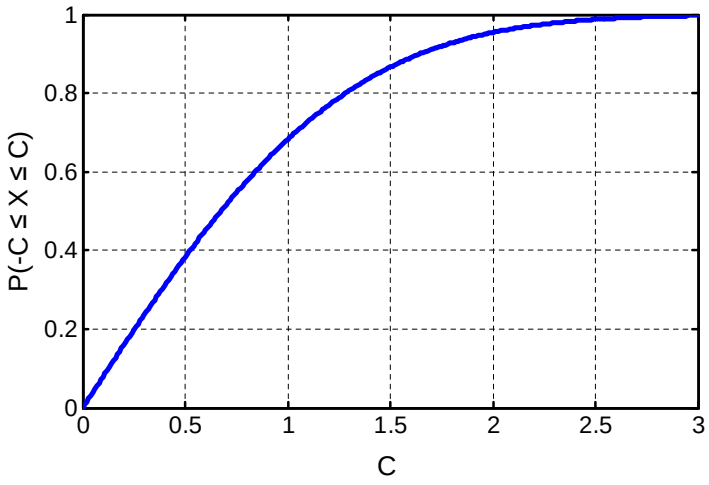
Gaussian Distribution

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



Yield (1)

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



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

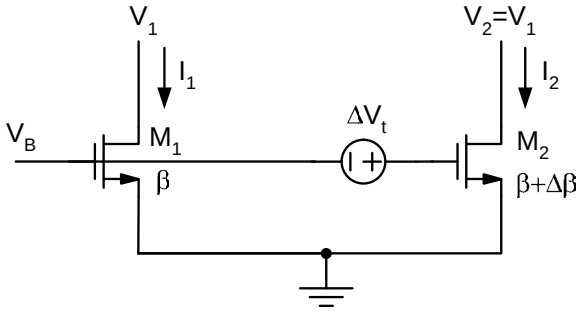
Example

- 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")

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

- 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

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

DNL of Thermometer DAC

$$DNL(k) = \frac{Step(k) - Step_{avg}}{Step_{avg}} \cong \frac{I_k - I}{I} = \frac{\Delta I}{I}$$

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

- 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
  - $\sigma_{DNL} = \sigma_u = 1\%$
  - $3\ \sigma_{DNL} = 3\%$
  - 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)
  - $0.9973^{1/63} = 0.99995708\ldots$
  - $0.9973^{1/4095} = 0.\ 99999929929\ldots$
- 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
- A more elaborate example is the topic of HW3

INL (1)

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

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

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

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

INL (1)

- For a quotient of random variables

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

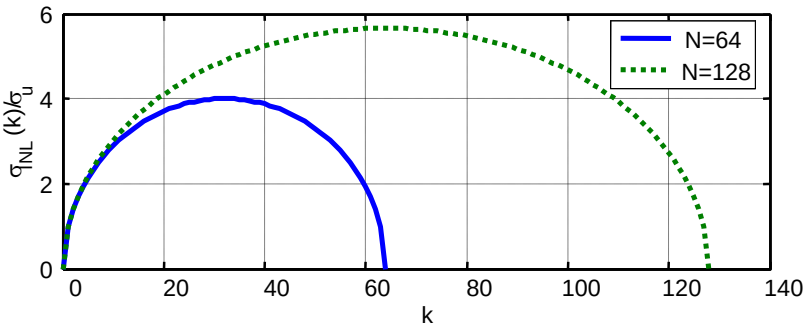
[Dennis E. Blumenfeld, *Operations Research Calculations Handbook*, Online: http://www.engnetbase.com/ejournals/books/book_summary/toc.asp?id=701]

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

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

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

INL (2)



- 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

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

- 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

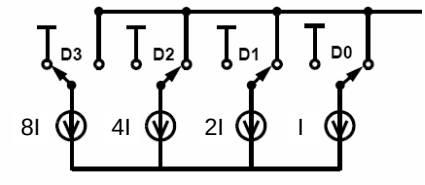
INL Yield

- 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

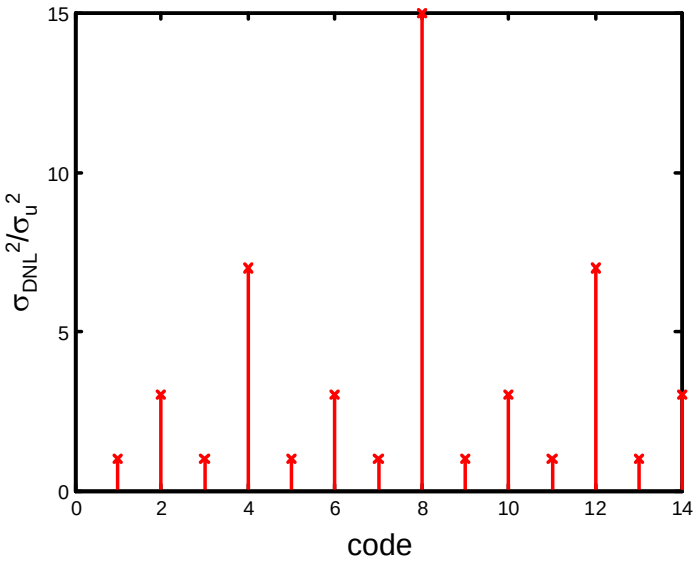
- INL same as for thermometer DAC
  - Why?
- DNL is not same for all codes, but depends on transition
- Consider worst case: 0111 … → 1000 …
  - Turning on MSB and turning off all LSBs

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

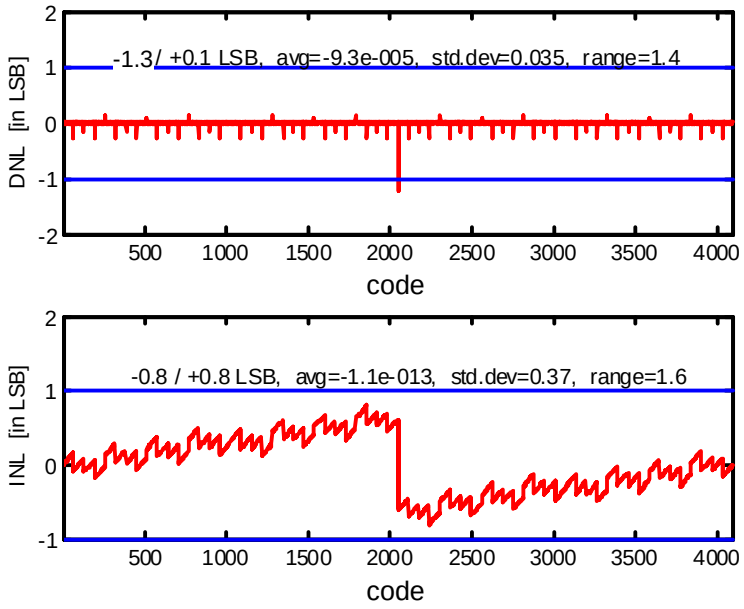


- Example
  - B = 12, σ_u = 1% → σ_{DNL} = 0.64 LSB
  - Much worse than thermometer DAC

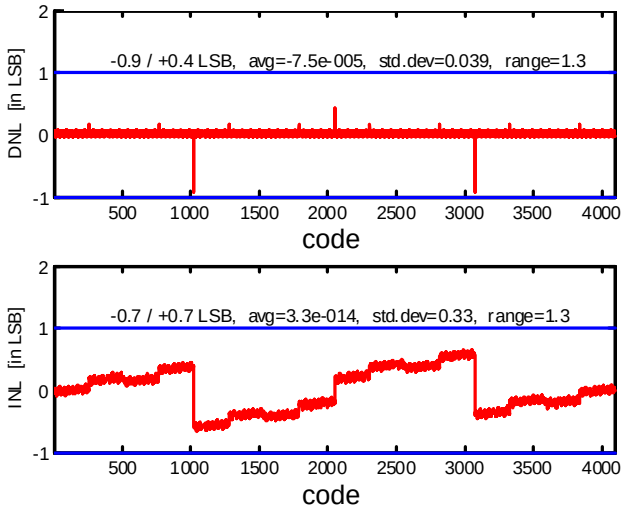
σ_{DNL} (4-bit Example)



Simulation Example



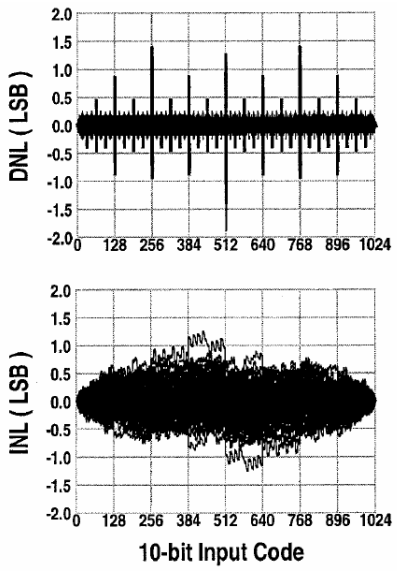
Another Random Run



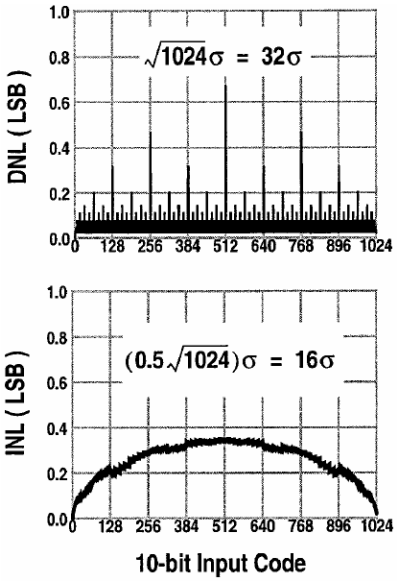
- Peak DNL not at mid-scale!
  - Important to realize that this is just one single statistical outcome…

Multiple Simulation Runs (100)

Overlay Plot



RMS DNL and INL

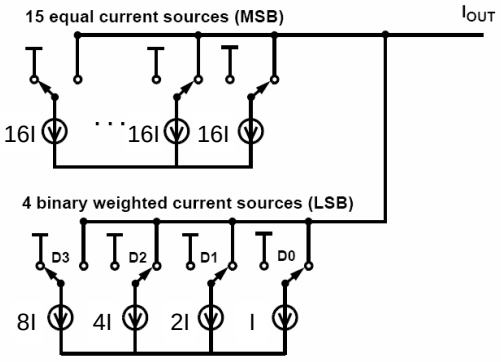


[Lin & Bult, JSSC 12/1998]

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

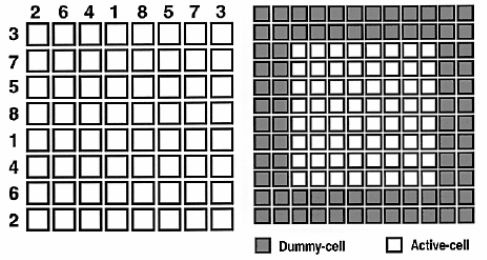
	Thermometer	Segmented	Binary Weighted
σ_{INL}	$\cong \frac{1}{2}\sigma_u\sqrt{2^B}$		
σ_{DNL}	$\cong \sigma_u$	$\cong \sigma_u\sqrt{2^{B_b+1}-1}$	$\cong \sigma_u\sqrt{2^B-1}$
Number of Switched Elements	2^B-1	$B_b+2^{B_t}-1$	B

Example (B=12, σ_u=1%)

DAC Architecture	σ_{INL}	σ_{DNL}	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

DAC INL/DNL Summary

- INL is independent of DAC architecture and requires element matching commensurate with overall DAC precision
- DAC architecture has significant impact on DNL
- Presented results are for uncorrelated random element variations
- Systematic errors and correlations are usually also important, but can be mitigated by proper layout and switching sequence design
  - See e.g. [Lin, JSSC 12/98], [Van der Plas, JSSC 12/99]



Dynamic DAC Errors (1)

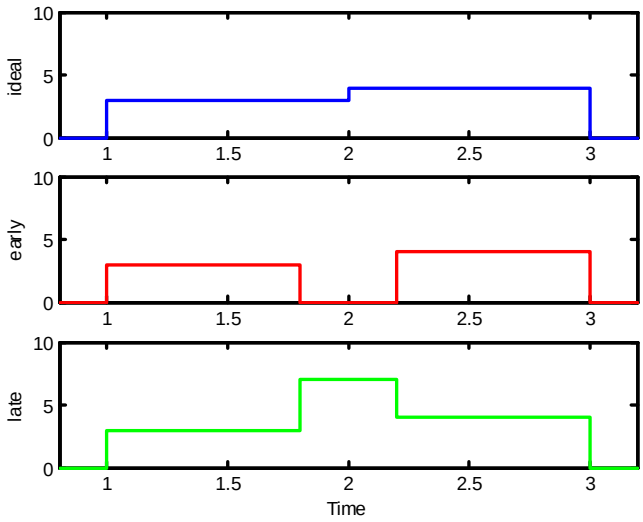
- Finite settling time and slewing
  - Finite RC time constant
  - Signal dependent slewing
- Feedthrough
  - Coupling from switch signals to DAC output
  - Clock feedthrough
- Glitches due to timing errors
  - Current sources won't switch simultaneously
- Dynamic DAC errors are generally hard to model!

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

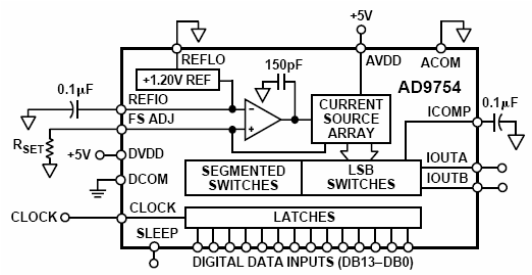
LSBs late, MSB early

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

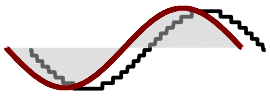


AD9754

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

Lecture 5
Nyquist Rate DACs (Continued)
Sampling Circuits



Boris Murmann
Stanford University
murmann@stanford.edu

DAC Example

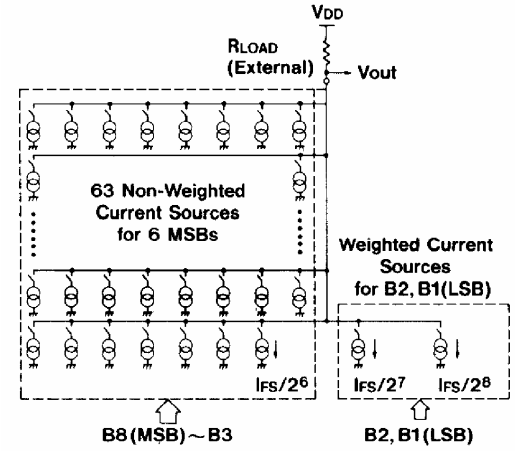


Fig. 1. Basic architecture of the DAC.

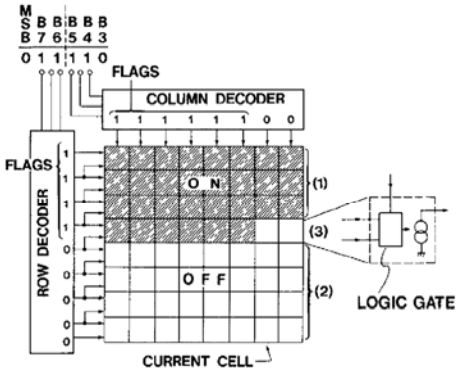
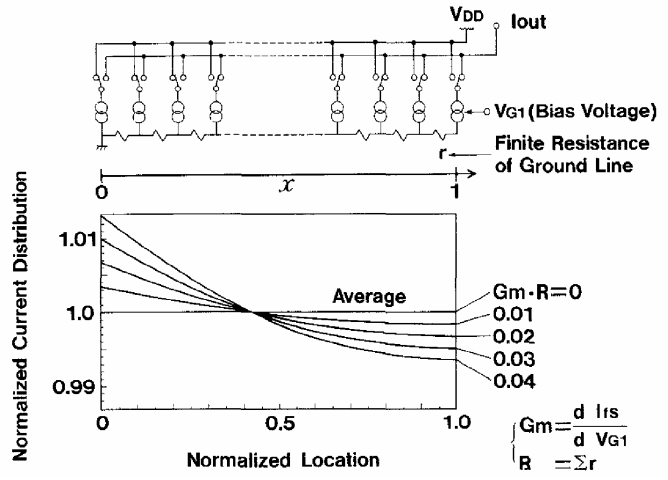


Fig. 2. Two-step decoding.

[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.]

Mitigating IR Drop



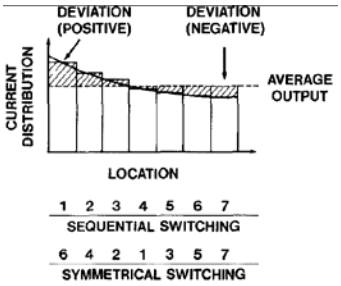
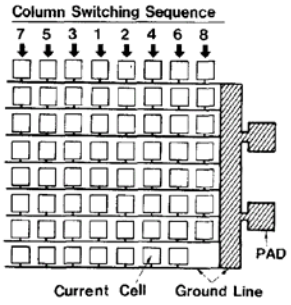
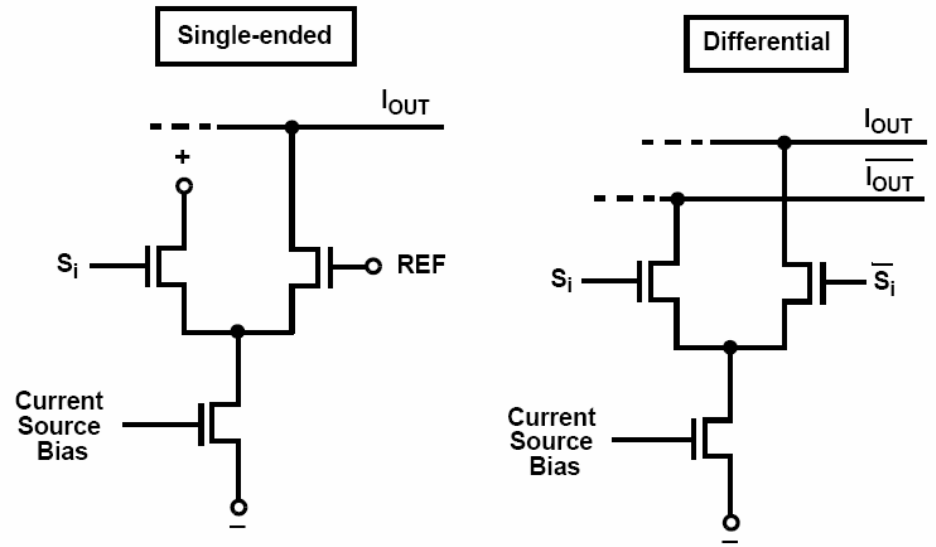


Fig. 9. Symmetrical switching.



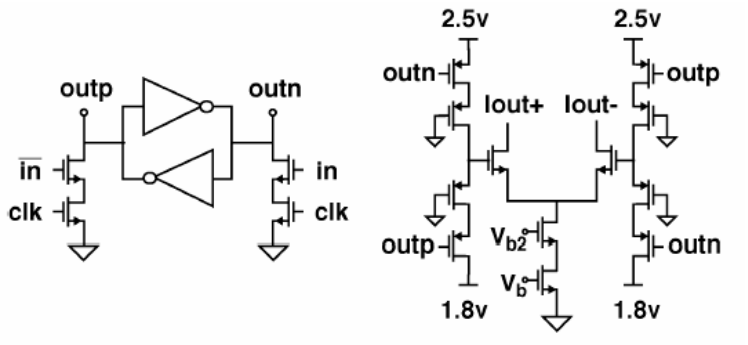
Basic Differential Pair Switch



Commonly Used Techniques

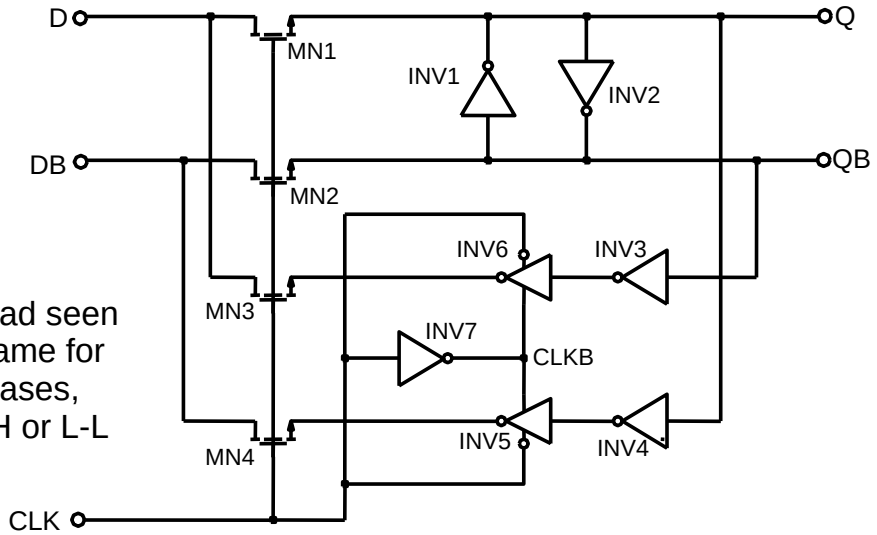
- Retiming
  - Latches in (or close to) each current cell
  - Latch controlled by global clock to ensure that current cells switch simultaneously (independent of decoder delays)
- Make before break
  - Ensure uninterrupted current flow, so that tail current source remains active
- Low swing driver
  - 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)

Example Current Cell Implementation



[Barkin & Wooley, JSSC 4/2004]

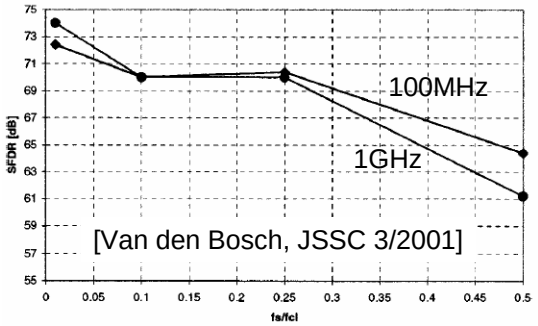
Constant Clock Load Latch



Capacitive load seen at CLK the same for all possible cases, H-L, L-H, H-H or L-L

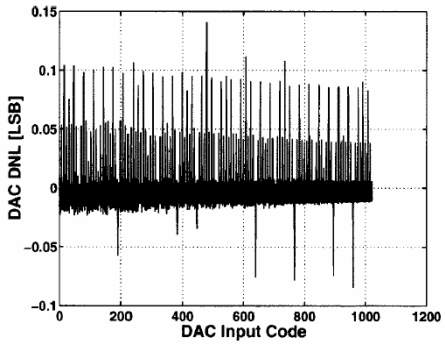
Mercer, *US patent ,7,023,255 4/4/2006*

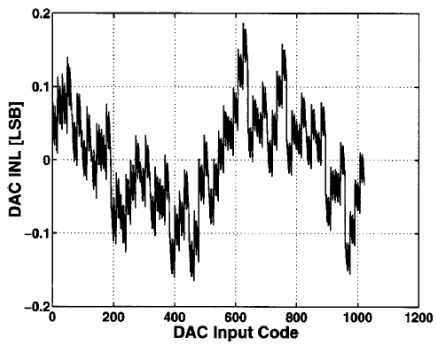
High Performance DAC Examples (1)



[Van den Bosch, JSSC 3/2001]

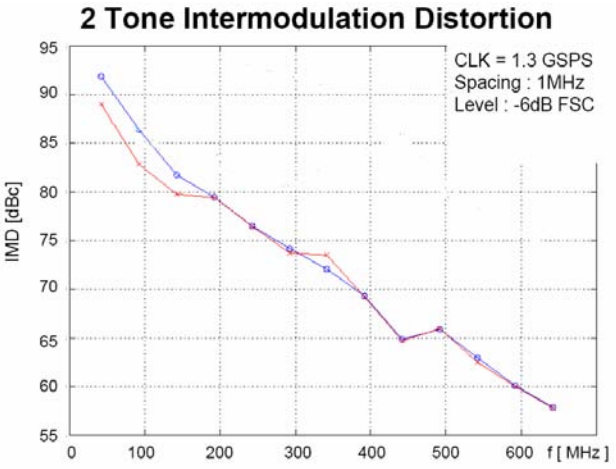
Resolution	10 bit
Update rate	Up to 1 GS/s
INL	< 0.2LSB
DNL	<0.15 LSB
SFDR (490MHz@1GS/s)	61.2 dB
Analog/digital voltage supply	3V / 1.9 V
Power consumption	110mW (490Mhz@1GHz)
Active area	0.35 mm^2
Process	0.35 µm CMOS





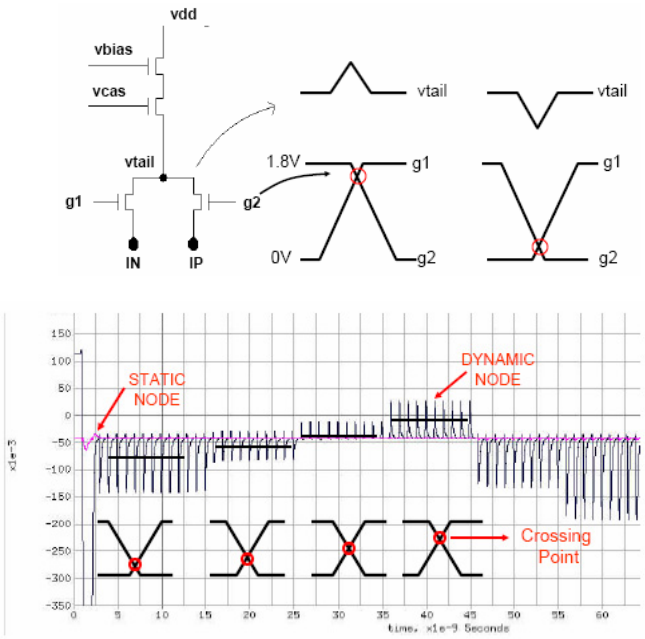
High Performance DAC Examples (2)

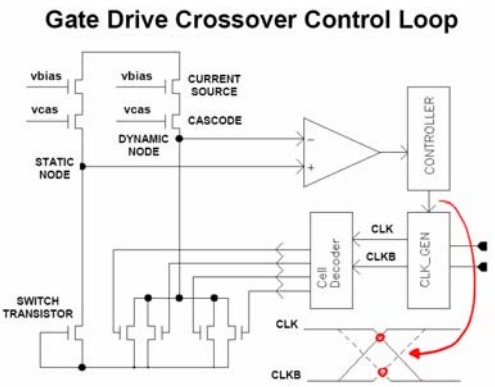
Max Sample Frequency	1.4	GSPS
Resolution	14	Bit
DNL	+/- 0.8	LSB
INL	+/- 2.1	LSB
SFDR @ 1.0 GSPS	> 60	dB
IMD @ 1.0 GSPS	> 64	dBc
NSD @ f_{out} = 400MHz	-155	dBm/Hz
Power (Core) @1.4GSPS	200	mW
Power(Total) @ 1.4GSPS	400	mW
Area (Core)	0.8	mm^2
Area (Chip)	6.25	mm^2



[Schafferer, ISSCC 2004]

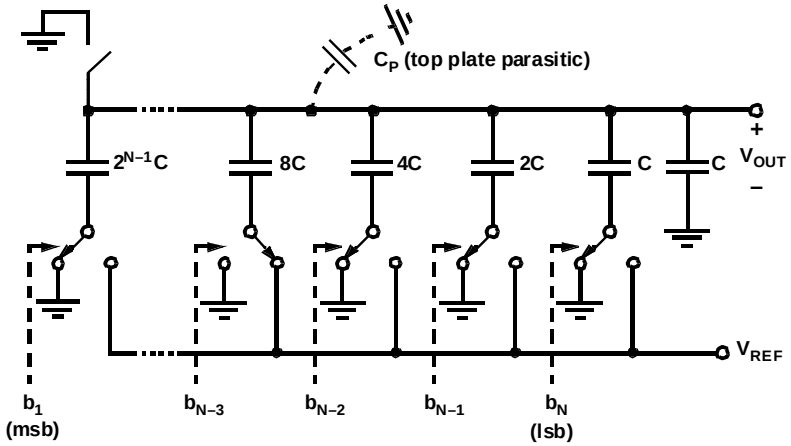
High Performance DAC Examples (3)





[Schafferer, ISSCC 2004]

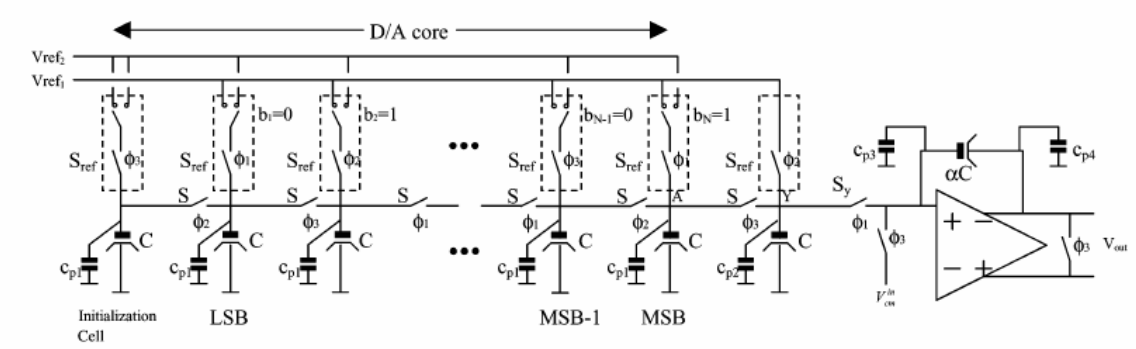
Binary Weighted Charge Redistribution DAC



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

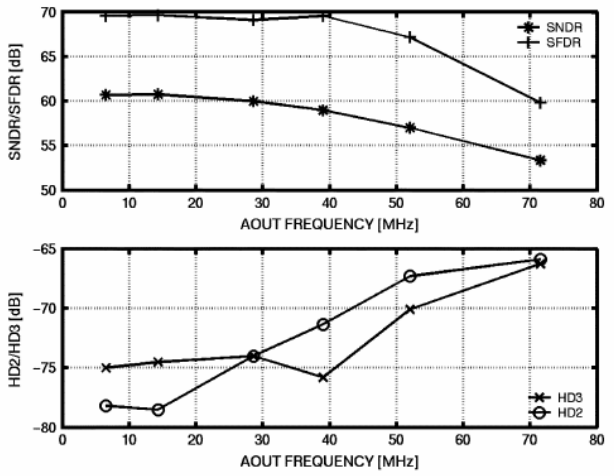
- Can redistribute charge onto OTA + feedback capacitor to mitigate gain error due to C_p

Charge-based Pipeline DAC (1)



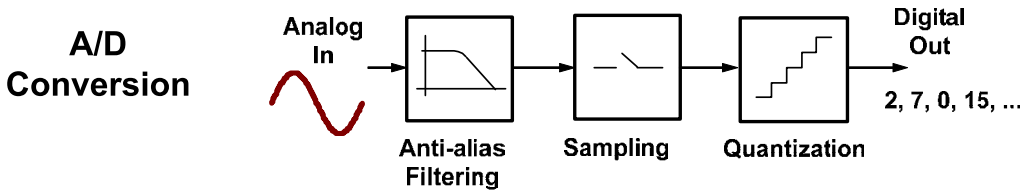
[Manganaro et al., "A dual 10-b 200-MSPS pipelined D/A converter with DLL-based clock synthesizer," IEEE JSSC 11/2004]

Charge-based Pipeline DAC (2)



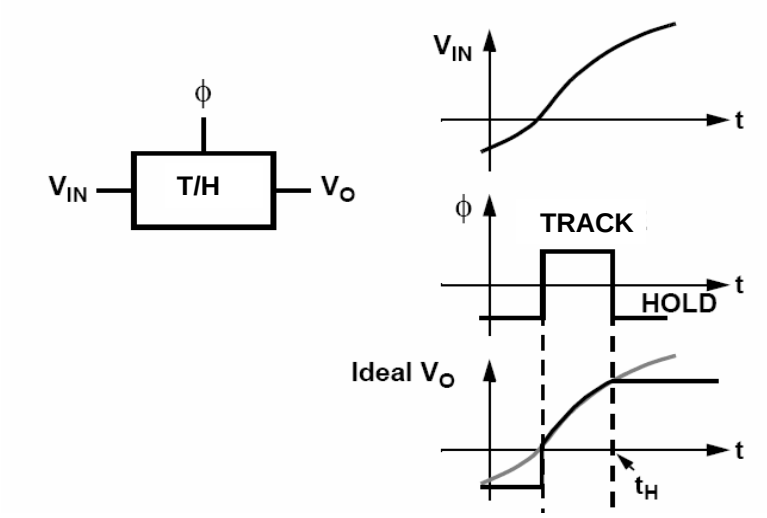
(f_{clk}=200MHz)

Recap



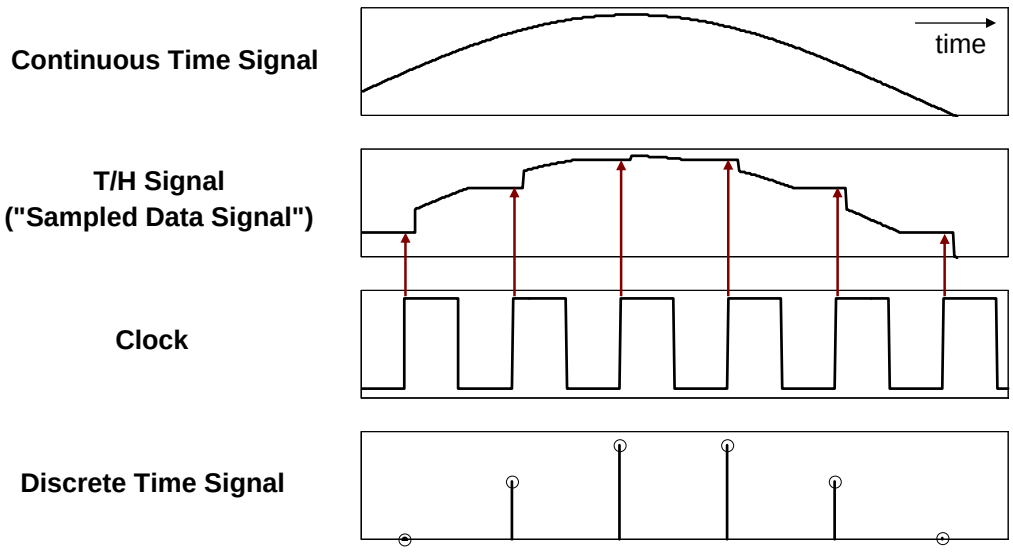
- How to build circuits that "sample"?
- Ideal Dirac sampling is impractical
  - Need a switch that opens, closes and acquires signal within an infinitely small time
- Practical solution
  - "Track and hold"

Ideal Track & Hold

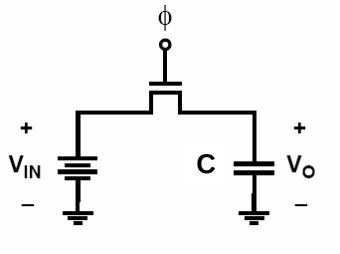


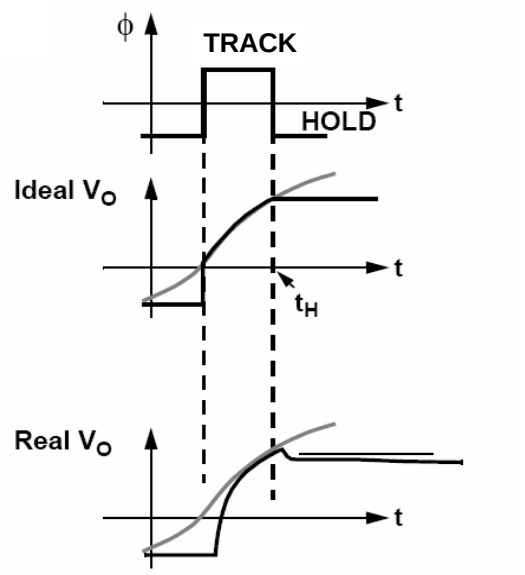
- Even though it's a somewhat inaccurate description, we sometimes call this circuit sample & hold…

Signal Nomenclature



Basic Track & Hold



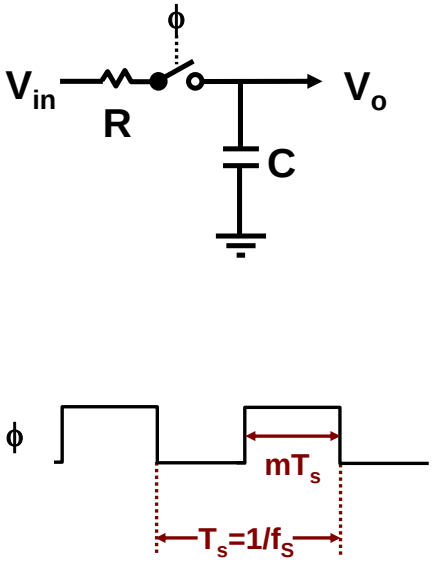


Overview

- Nonidealities
  - Finite acquisition time
  - kT/C noise
  - Aperture uncertainty
  - Signal dependent sampling instant
  - Hold mode feedthrough and droop
  - Track mode nonlinearity, R = f(V_{in})
  - Pedestal error, charge injection
- Compensation for nonidealities
  - CMOS switch, clock boosting
  - Dummy switch
  - Fully differential bottom plate sampling

Finite Acquisition Time (1)

- Finite speed in track mode due to time constant $\tau = RC$
- What are the constraints on τ for a given sampling rate and resolution ?
- Consider following example
  - Switch open, $V_o=0$
  - Switch closes with constant $V_{in} = V_{FS}$ applied
  - Calculate required τ such that V_{out} settles to within fraction of LSB within mT_s
    - Usually $m \cong 0.5$



Finite Acquisition Time (2)

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

$$V_{FS} \cdot e^{-mT_s/\tau} < \alpha\Delta$$

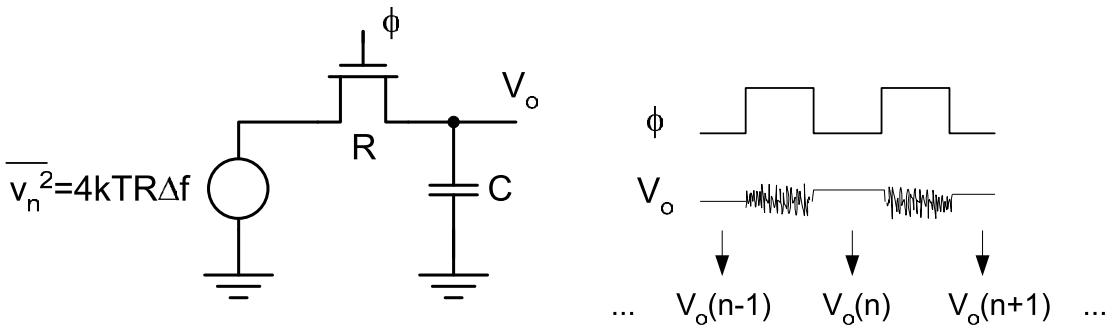
$$2^B \Delta \cdot e^{-mT_s/\tau} < \alpha\Delta$$

$$M = \frac{mT_s}{\tau} > ln\left(\frac{2^B}{\alpha}\right)$$

"Number of settling time constants"

B	M (α=0.5)
6	4.9
10	7.6
14	10.4
18	13.2

Thermal Noise (1)



- Questions
  - What is the noise variance of the V_o samples in hold mode?
  - What is the spectrum of the discrete time sequence representing these samples?

Thermal Noise (2)

- Sample values $V_o(n)$ correspond to instantaneous values of the track mode noise process
- From Parseval's theorem, we know that the time domain power (or variance) of this process is equal to its power spectral density integrated over all frequencies
  - Further, given that the process is ergodic, this number must also be equal to the "ensemble" variance, i.e. the variance of a sample taken at a particular time

$$\frac{\overline{v_o^2}}{\Delta f} = 4kTR \cdot \left| \frac{1}{1+sRC} \right|^2$$

$$var[V_o(n)] = \overline{v_{o,tot}^2} = \int_0^\infty 4kTR \cdot \left| \frac{1}{1+j2\pi f \cdot RC} \right|^2 df = \frac{kT}{C}$$

Alternative Derivation

- The equipartition theorem (statistical mechanics) says that each "quadratic degree of freedom" of a system in thermal equilibrium holds an average energy of kT/2
  - See e.g. EEAP248 for a derivation
- In our system, the quadratic degree of freedom is the energy stored on the capacitor

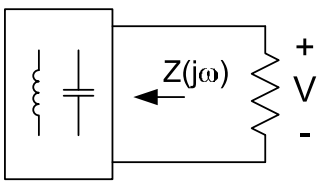
$$\overline{\frac{1}{2}Cv_o^{\,2}} = \frac{1}{2}kT$$

$$\overline{v_o^{\,2}} = \frac{kT}{C}$$

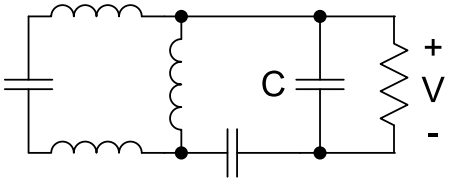
Another Interesting Theorem

- Consider the parallel connection of a resistor and an arbitrary (passive) reactive network with port impedance Z(jω)

$$\frac{1}{C} = \lim_{\omega\to\infty} j\omega Z(j\omega) \qquad \Rightarrow \overline{v_{tot}^2} = \frac{kT}{C}$$



- For a proof see
  - Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd ed., pp. 352, McGraw Hill.
- Example



$$\Rightarrow \overline{v_{tot}^2} = \frac{kT}{C}$$

Implications of kT/C Noise

- If we make kT/C noise equal to quantization noise

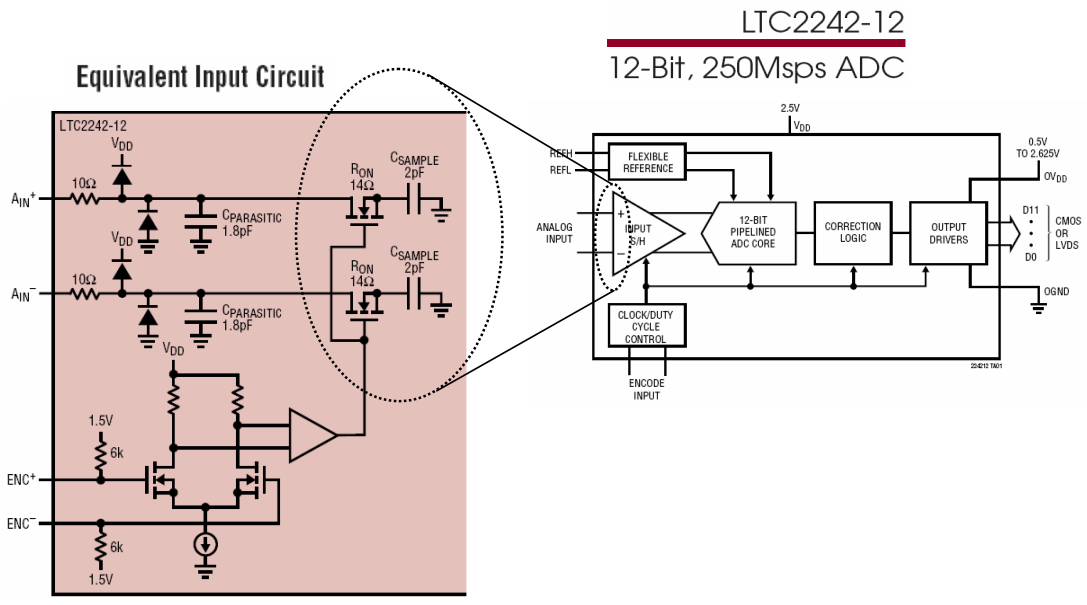
$$\frac{kT}{C} = \frac{\Delta^2}{12} \quad \Rightarrow \quad C = 12kT\left(\frac{2^B}{V_{FS}}\right)^2$$

- Example RC values using this assumption and V_{FS}=1V, α=0.5, m=0.5, f_s=100MHz

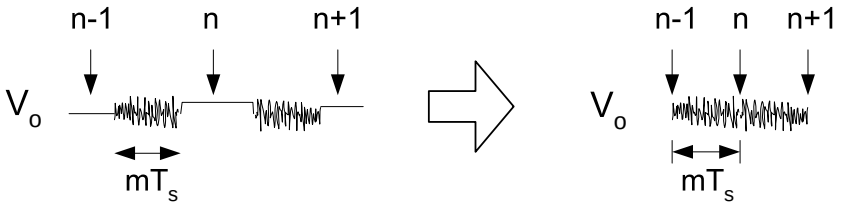
B	C [pF]	R [Ω]
8	0.003	246,057
10	0.052	12,582
12	0.834	665
14	13.3	36
16	213	1.99
18	3,416	0.11

- Oversampling helps reduce capacitor sizes (more later in this class)
  - Especially useful at high resolution

Commercial Example



Spectrum of Noise Samples



- Strategy
  - Realize that discrete time noise samples are essentially instantaneous values (mT_s apart) of the continuous time noise process in track mode
  - Spectrum follows from Fourier transform of the process' autocorrelation function (Wiener-Khintchin)
    - Samples show no correlation → white spectrum
    - Samples are correlated → colored spectrum

Analysis (1)

- Calculate autocorrelation function

Autocorrelation of resistor noise (white) | Impulse response of RC filter | Autocorrelation of filtered noise

$$R_{xx}(\tau) = \delta(0) \cdot 2kTR \longrightarrow \boxed{h(t) = \frac{1}{RC} e^{-t/RC}} \longrightarrow R_{yy}(\tau) = R_{xx}(\tau) * h(\tau) * h(-\tau)$$

$$\therefore R_{yy}(\tau) = \frac{kT}{C} e^{-\frac{|\tau|}{RC}}$$

$$\therefore R_{yy}(k) = \frac{kT}{C} e^{-\frac{|k \cdot mT_s|}{RC}}$$

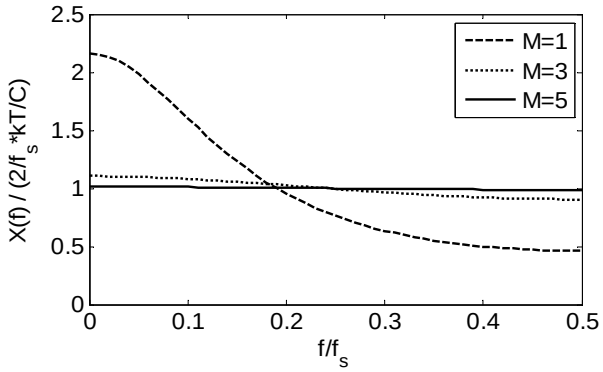
Covariance of samples separated by k clock cycles

Analysis (2)

- Apply discrete time Fourier transform

$$X(\omega) = \frac{kT}{C}\sum_{-\infty}^{\infty} R_{yy}(k) e^{j\omega \cdot kT_s}$$

$$X(f) = \frac{2}{f_s}\frac{kT}{C}\frac{1-e^{-2M}}{1-2e^{-M}\cos\left(2\pi\frac{f}{f_s}\right)+e^{-2M}} \qquad M = \frac{mT_s}{RC}$$



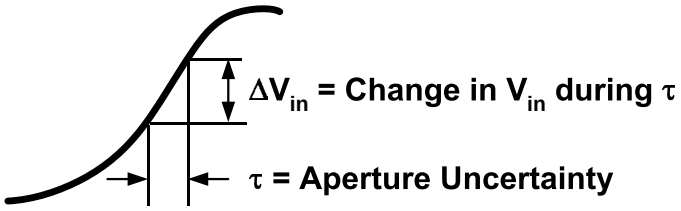
Spectrum of noise samples is essentially white for M>3

Makes intuitive sense

– Large M means that noise sample values decay from one cycle to next

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



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

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

Analysis

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

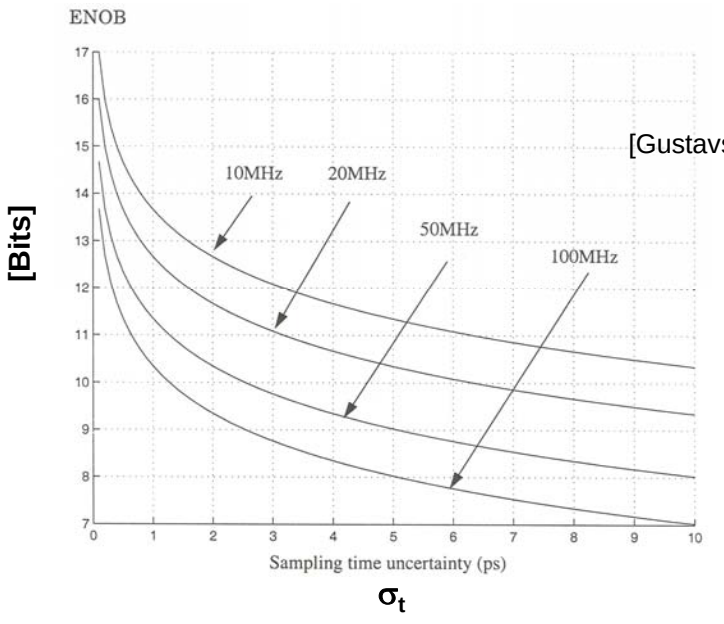
$$E\left\{\Delta V_{in}^2\right\} \cong E\left\{\left(\frac{dV_{in}}{dt}\right)^2 \cdot \tau^2\right\} = E\left\{\left(\frac{dV_{in}}{dt}\right)^2\right\} \cdot E\left\{\tau^2\right\}$$

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

$$\cong \frac{1}{2}\left(2\pi \cdot A \cdot f_{in}\right)^2 \cdot \sigma_t^2$$

$$\boxed{SNR_{aperture} \cong 10 \cdot \log \frac{1}{\left(2\pi \cdot f_{in} \cdot \sigma_t\right)^2}}$$

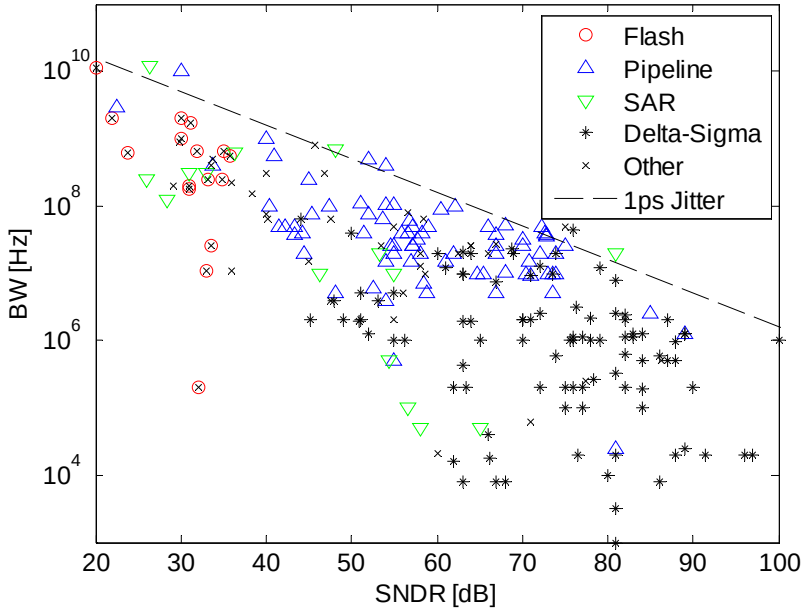
Result



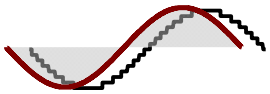
[Gustavsson, p.63]

ADC Performance Survey (ISSCC & VLSI 97-08)

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



Lecture 6
Sampling Circuits (Continued)



Boris Murmann
Stanford University
murmann@stanford.edu

Voltage Dependence of Switch

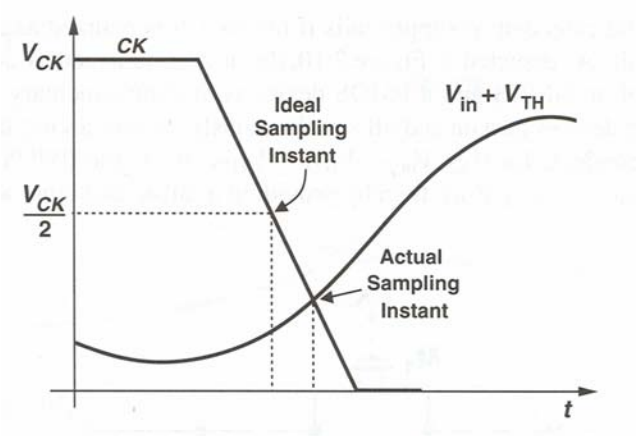
$$I_{D(triode)} = \mu C_{ox} \frac{W}{L}\left(V_{GS} - V_t - \frac{V_{DS}}{2}\right)V_{DS}$$

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

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

- 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.)

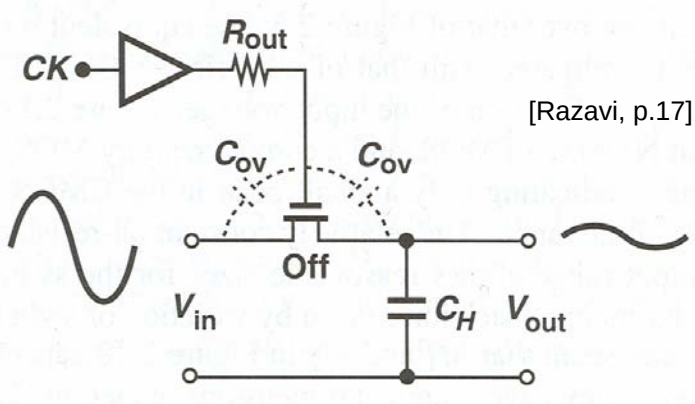
Signal Dependent Sampling Instant



[Razavi, p.17]

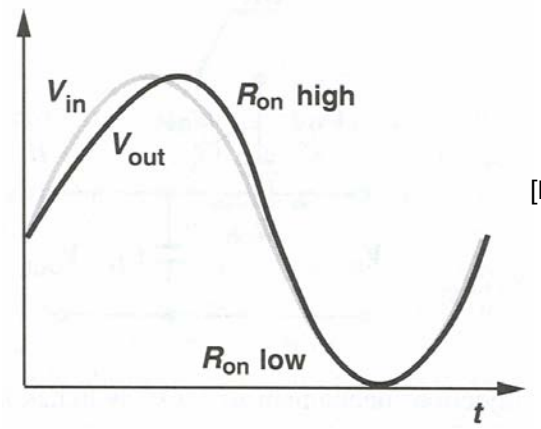
- Must make fall time of sampling clock much faster than maximum dV_{in}/dt

Hold Mode Feedthrough



[Razavi, p.17]

Track Mode Nonlinearity



[Razavi, p.16]

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

Analysis

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

$$C\frac{dV_o}{dt} = K(\phi - V_o - V_t)(V_i - V_o) - \frac{K}{2}(V_i - V_o)^2$$

- "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
- Luckily someone has already done this for us
  - W. Yu *et al.*, "Distortion analysis of MOS track-and-hold sampling mixers using time-varying Volterra series," *IEEE Trans. Ckts. Syst. II*, pp. 101-113, Feb. 1999.

Result

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

$$\cong \frac{1}{4}\frac{A^2}{(V_{GS}-V_t)^2}\cdot\frac{2\pi\cdot f_{in}\cdot C}{K(V_{GS}-V_t)}$$

$$\cong \frac{1}{4}\frac{A^2}{(V_{GS}-V_t)^2}\cdot 2\pi\cdot f_{in}\cdot RC$$

- Here, R and V_{GS} are the respective "quiescent point" values
- For low distortion
  - Make amplitude smaller than V_{GS}-V_t
    - Low swing
  - Make 1/RC much larger than $2\pi\cdot f_{in}$
    - Big switch

Example ($f_{in}=f_s/2$)

$$|HD_3| \cong \frac{1}{4}\frac{A^2}{(V_{GS}-V_t)^2}\cdot 2\pi\cdot f_{in}\cdot RC$$

$$\cong \frac{\pi}{4}\frac{A^2}{(V_{GS}-V_t)^2}\cdot\frac{\tau}{T_s}$$

- Assumptions
  - Signal is centered about $V_{DD}/2$=0.9V
  - V_{GS}-V_t = 1.8V-0.9V-0.45V = 0.45V, A=0.2V
  - T_s/τ = 20

$$\therefore |HD_3| \cong \frac{\pi}{4}\frac{0.2^2}{0.45^2}\cdot\frac{1}{20} = -42dB$$

Measured Data

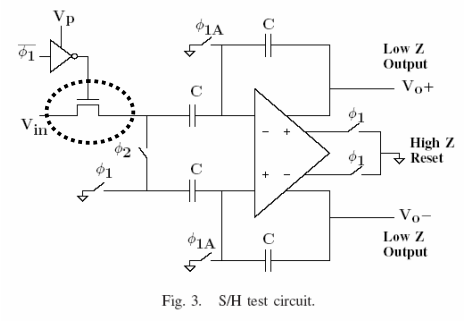


Fig. 3. S/H test circuit.

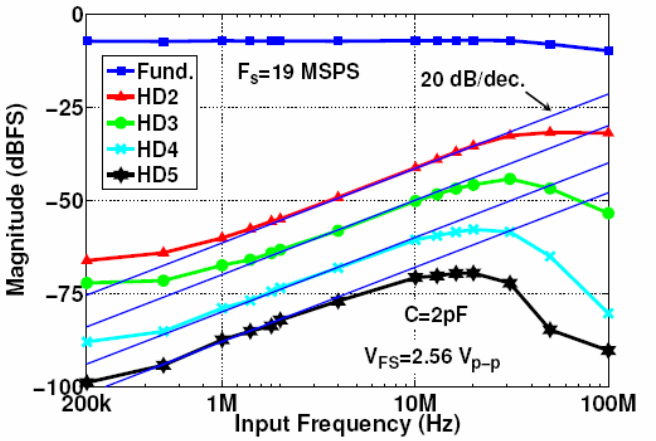
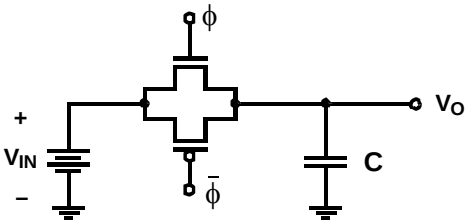


Fig. 5. Measured HD vs. f_{in}, F_s = 19MSPS.

[Brown et al., "Prediction and Characterization of Frequency Dependent MOS Switch Linearity and the Design Implications," CICC 2006]

CMOS Switch (1)



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

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

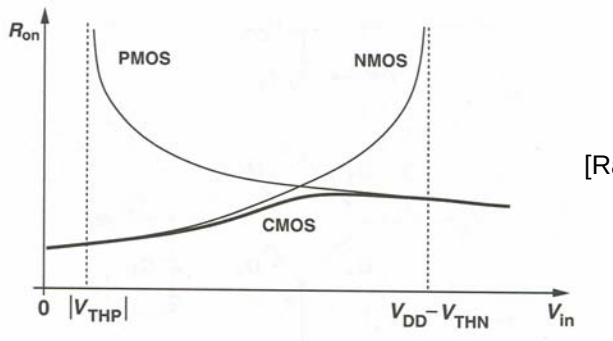
CMOS Switch (2)

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

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

- Independent of V_{in} - too good to be true…
- Missing factors
  - Back-gate effect
  - Short channel effects

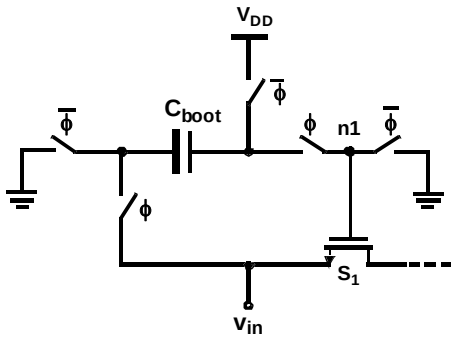
Real CMOS Switch



[Razavi, p.19]

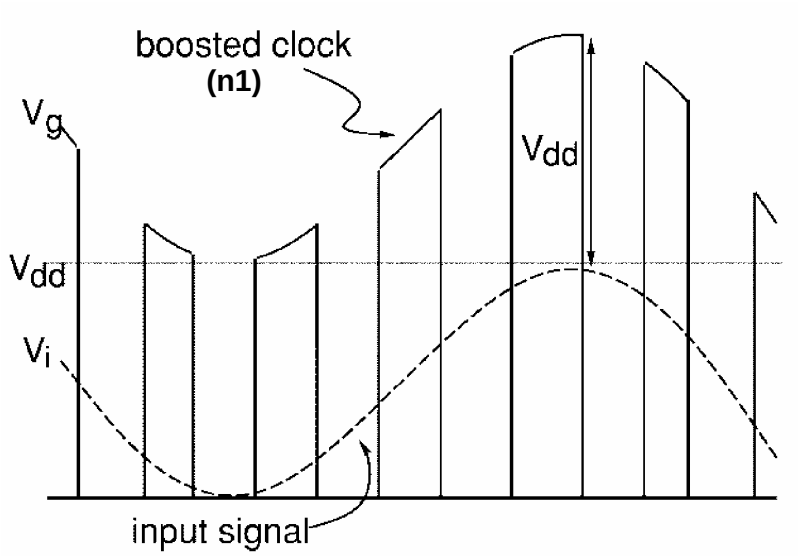
- Design
  - Size P/N device ratio to minimize change in R_{ON} over desired input range
  - Size P and N simultaneously to meet distortion specs
- Remaining issue
  - P and N turn off at slightly different times

Clock Bootstrapping

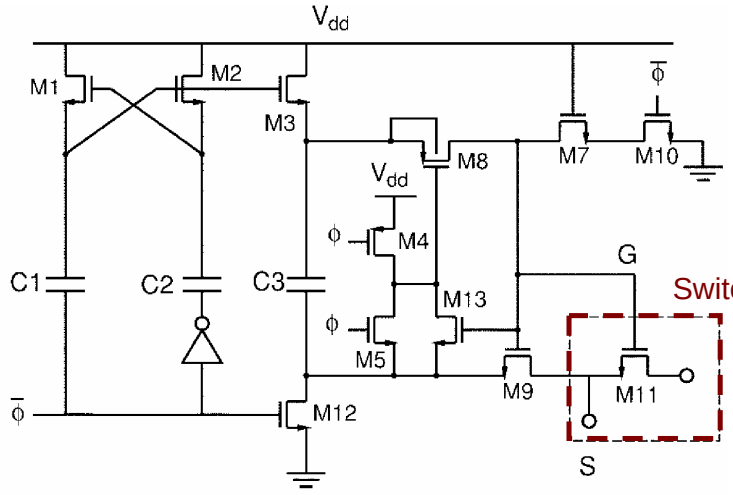


- ϕ LOW
  - C_{boot} is precharged to V_{DD}
  - Sampling switch is off
- ϕ HIGH
  - Constant voltage, equal to V_{DD} is established between gate and source terminal of sampling switch

Waveforms



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 Ckts., pp. 599, May 1999]

Limitations

- Ideally switch on-resistance is independent of input signal
- In practice, parasitic capacitance at gate node (n1) and body effect limit achievable linearity

$$R \cong \frac{1}{\mu_n C_{ox}\left[\frac{W}{L}\right]_n\left(\frac{C_{boot}}{C_{boot}+C_{n1}}V_{DD}-\frac{C_{n1}}{C_{boot}+C_{n1}}V_{in}-V_{tn}[V_{in}]\right)}$$

Alternative Implementation

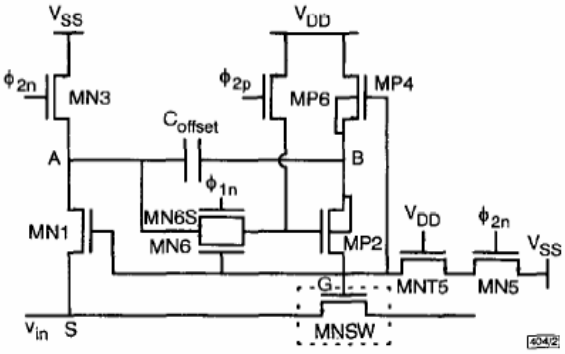
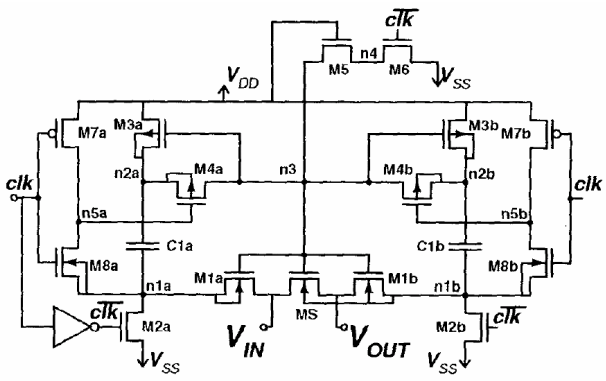


Fig. 2 *Proposed implementation*

Dessouky & Kaiser, "Input switch configuration suitable for rail-to-rail operation of switched opamp circuits," Electronics Letters, Jan. 1999]

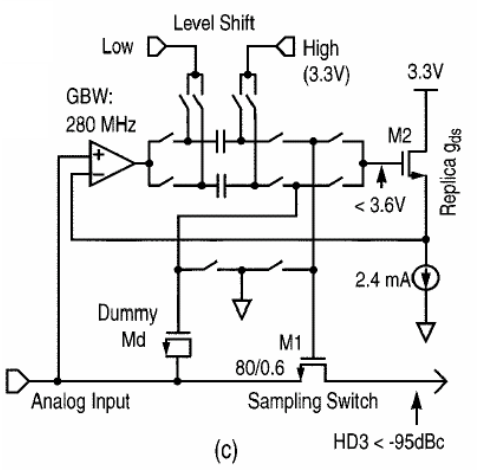
Advanced Clock Boostrapping (1)



- Gate tracks average of input and output, reduces effect of I·R drop at high frequencies
- Bulk also tracks signal
  - Reduced body effect
- Measured SFDR = 76.5dB at f_{in}=200MHz

[M. Waltari et al., "A self-calibrated pipeline ADC with 200MHz IF-sampling front-end," ISSCC 2002, Dig. Techn. Papers, pp. 314.]

Advanced Clock Boostrapping (2)

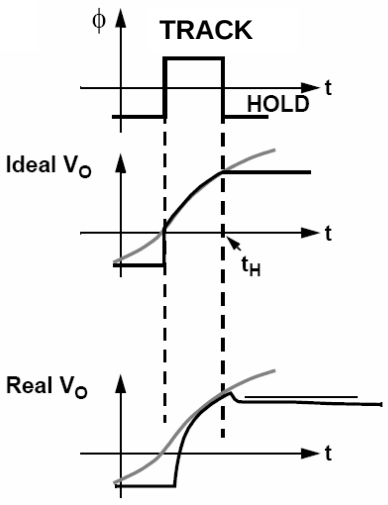


- An attempt to cancel body effect

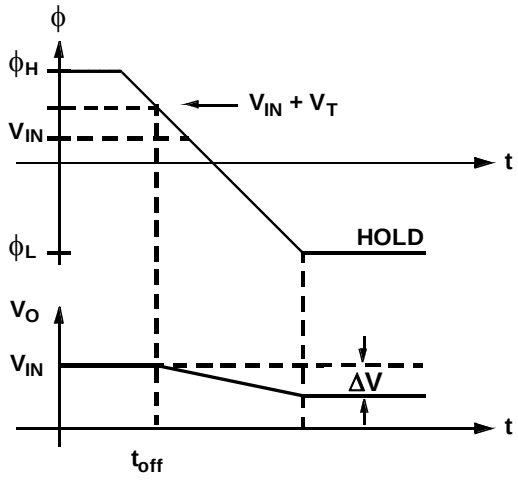
[H. Pan et al., "A 3.3-V 12-b 50-MS/s A/D converter in 0.6um CMOS with over 80-dB SFDR," *IEEE J. Solid-State Circuits*, pp. 1769-1780, Dec. 2000]

Pedestal Error

- Error introduced at the output during transition from track to hold
- Caused by charge injection
  - Charge from overlap capacitance and channel
- Depends on clock transition time (waveform of ϕ)
- Two interesting cases
  - "Quasi static gating" ("slow gating")
  - "Fast gating"

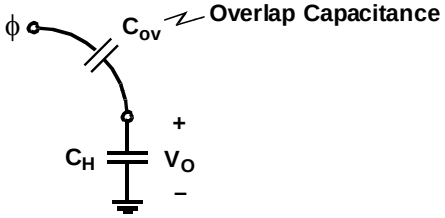


Slow Gating



- Channel charge has disappeared by t_{off} without introducing error
  - All channel charge absorbed by input source

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



$$\Delta V = \frac{C_{OV}}{C_{OV}+C_H}\left(V_{in}+V_t-\phi_L\right)$$

$$V_0 = V_{in}-\Delta V = V_{in}-\frac{C_{OV}}{C_{OV}+C_H}\left(V_{in}+V_t-\phi_L\right)$$

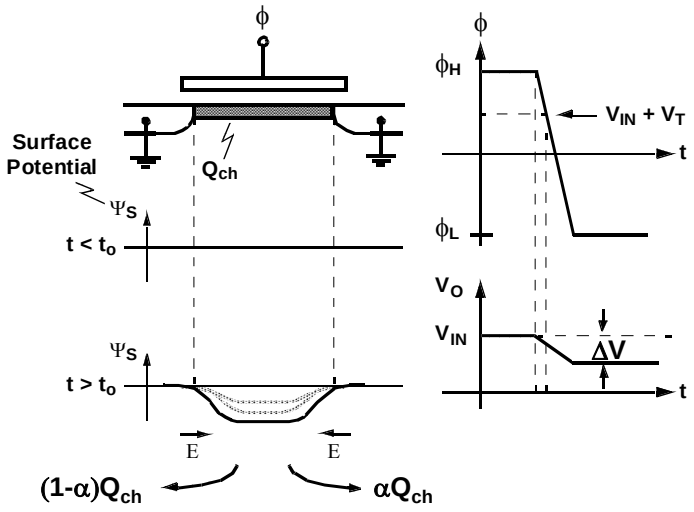
$$V_0 = V_{in}\left(1+\varepsilon\right)+V_{os}$$

$$\varepsilon = -\frac{C_{OV}}{C_{OV}+C_H}$$

$$V_{os} = -\frac{C_{OV}}{C_{OV}+C_H}\left(V_t-\phi_L\right)$$

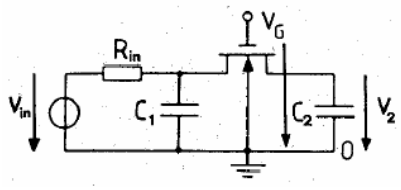
- Example
  - C_H=1pF, ϕ_L=0V, V_t=0.45V, W=20μm, C_{GD0}=0.1fF/μm ⇒ C_{OV}=2fF
  - ε=-0.2%, V_{os}=-0.9mV

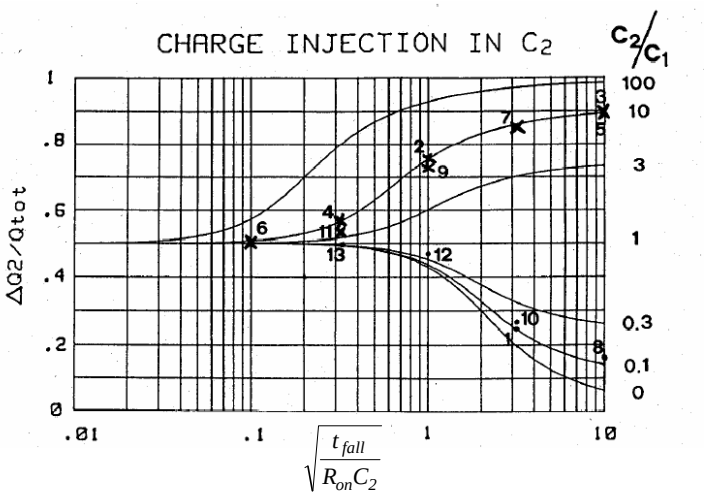
Fast Gating



- 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

Charge Split Ratio





[G. Wegmann *et al.*, "Charge injection in analog MOS switches," *IEEE J. of Solid-State Circuits*, pp. 1091-1097, June 1987]

[Y. Ding and R. Harjani, "A universal analytic charge injection model," *Proc. ISCAS*, pp. 144-147, 2000]

Fast Gating Model ($t>t_{off}$)

$$\Delta V = \frac{C_{OV}}{C_{OV}+C_H}(\phi_H - \phi_L) + \frac{1}{2}\frac{Q_{ch}}{C_H}$$

(assuming equal charge split for simplicity)

$$Q_{ch} = C_{ox}WL_{elec}[\phi_H - V_{in} - V_t]$$

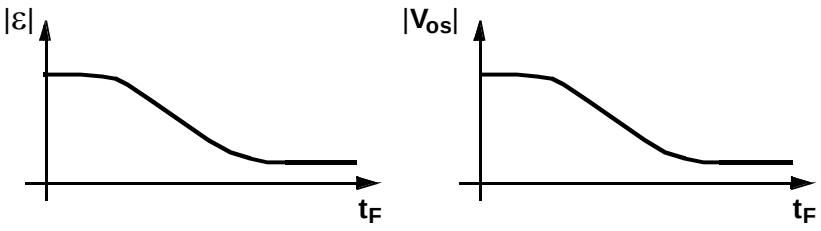
$$V_0 = V_{in} - \Delta V = V_{in}(1+\varepsilon) + V_{os}$$

$$\varepsilon = \frac{1}{2}\frac{C_{ox}WL_{elec}}{C_H}$$

$$V_{os} = -\frac{C_{OV}}{C_{OV}+C_H}(\phi_H - \phi_L) - \frac{1}{2}\frac{C_{ox}WL_{elec}}{C_H}(\phi_H - V_t)$$

- Example
  - C_H=1pF, ϕ_H=1.8V, ϕ_L=0V, V_t=0.45V, W=20μm, $C_{ox}L_{elec}$=2fF/μm C_{GD0}=0.1fF/μm ⇒ C_{OV}=2fF
  - ε=2%, V_{os}= -3.6mV-27mV = -30.6mV

Transition Fast/Slow Gating



- |ε| and $|V_{os}|$ decrease as the fall time of ϕ increases and approach the limit case of slow gating

Fundamental Speed/Accuracy Tradeoff

$$\Delta V \cong \frac{1}{2}\frac{Q_{ch}}{C} \qquad \frac{1}{2f_s} = \frac{T_s}{2} = M \cdot RC$$

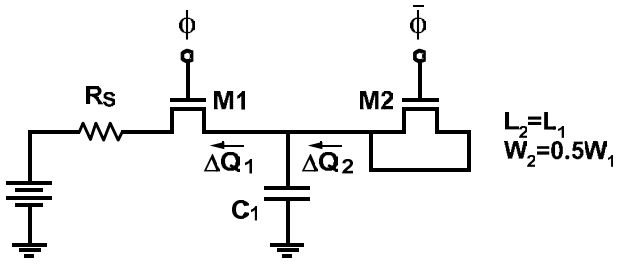
$$R \cong \frac{1}{\mu C_{ox}\frac{W}{L}(V_{GS} - V_t)} = \frac{L^2}{\mu Q_{ch}}$$

$$\boxed{\frac{\Delta V}{f_s} \cong M\frac{L^2}{\mu}}$$

- Example

$$10\frac{(0.18\,\mu m)^2}{400\frac{cm^2}{Vs}} \cong 8\frac{mV}{GHz}$$

Charge Cancellation

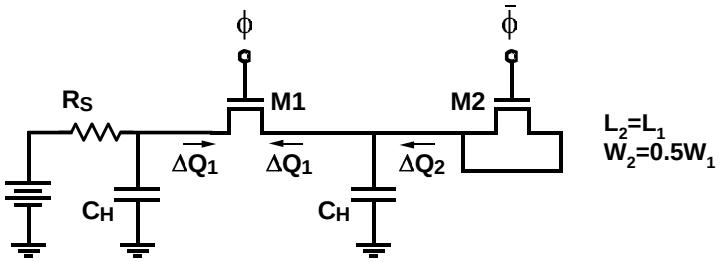


$$\Delta Q_1 \cong \frac{1}{2}Q_{ch1} + Q_{ov1}$$

$$\Delta Q_2 \cong Q_{ch2} + 2 \cdot Q_{ov2} \cong \frac{1}{2}Q_{ch1} + Q_{ov1}$$

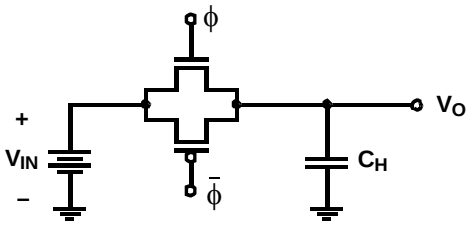
- See e.g. Eichenberger & Guggenbűhl, JSSC 8/89
- Can use dummy switch to inject charge packet with opposite sign
- Cancellation is never perfect, since channel charge of M1 will not exactly split 50/50
  - E.g. if R_s is very small, most of the charge will flow toward the input voltage source

Equalization Capacitor



- Bienstman & De Man, JSSC 12/80
- Much better cancellation
- Issue: Reduced bandwidth

CMOS Switch



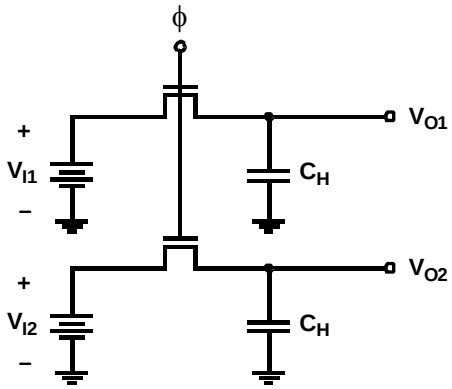
$$\frac{1}{2}|Q_{chn}| \cong \frac{1}{2}W_n L_n C_{ox}(\phi_H - V_{IN} - V_{tn})$$

$$\frac{1}{2}|Q_{chp}| \cong \frac{1}{2}W_p L_p C_{ox}(V_{IN} - \phi_L - |V_{tp}|)$$

Fast gating: $$\Delta V_O \cong \frac{-\frac{1}{2}Q_{chn} + \frac{1}{2}Q_{chp}}{C_H}$$

- Can achieve partial cancellation
- Issue: cancellation is signal dependent
- Want $W_nL_n = W_pL_p$
  - May not be so great for good tracking linearity and high speed (may want to use $L_p=L_n=L_{min}$, $W_p=2\ldots3\cdot W_n$)

Differential Sampling (1)



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

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

$$V_{O1} = (1 + \varepsilon_1)V_{I1} + V_{OS1}$$
$$V_{O2} = (1 + \varepsilon_2)V_{I2} + V_{OS2}$$

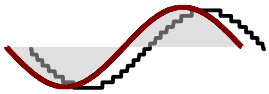
$$V_{OD} = \left(1 + \frac{\varepsilon_1 + \varepsilon_2}{2}\right)V_{ID} + (\varepsilon_1 - \varepsilon_2)V_{IC} + (V_{OS1} - V_{OS2})$$

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

Differential Sampling (2)

- Assuming good matching between half circuits
  - Only 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
- Other headaches
  - In addition to the linear gain error we considered, there will also be nonlinear terms (body effect, …)
  - Second order terms will cancel, but third order terms won't
    - Limits achievable HD_3, SFDR
- Solution: "bottom plate sampling"
  - More later…

Lecture 7
Sampling Circuits (Continued)

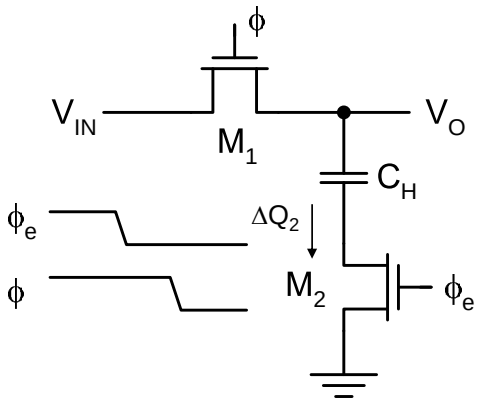


Boris Murmann
Stanford University
murmann@stanford.edu

Bottom Plate Sampling (1)

- Basic idea
  - Sample signal at the "grounded" side of the capacitor to achieve signal independence
- 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 (2)



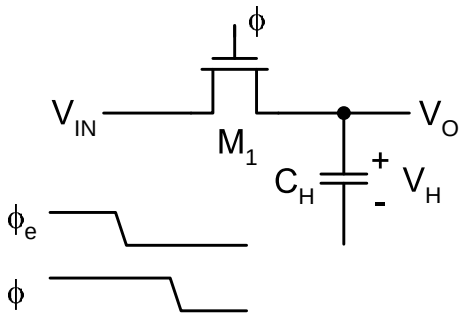
- Turn M_2 off "slightly" before M_1
  - Typically a few hundred ps delay between falling edges of ϕ_e and ϕ
- During turn off, M_2 injects charge

$$\Delta Q_2 \cong \alpha_2 WLC_{ox}(\phi_H - V_{tn})$$

- To first order, charge injected by M_2 is signal independent!
- Voltage across C_H

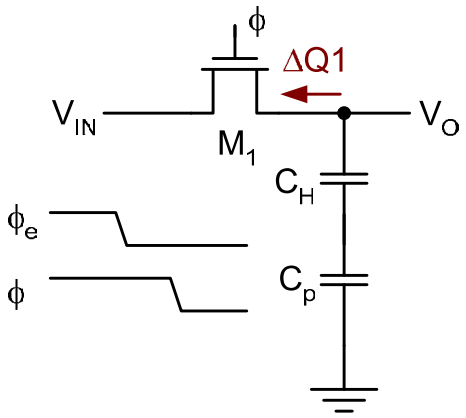
$$V_H = V_{IN} + \frac{\Delta Q_2}{C_H}$$

Bottom Plate Sampling (3)



- Next, turn off M1
- Since bottom plate of C_H is floating, there is no way to change its stored charge
  - M1 cannot inject any charge onto C_H
  - Most of M_1's charge injection goes to input source and/or onto parasitics at node V_O
- But, is the bottom plate really floating?

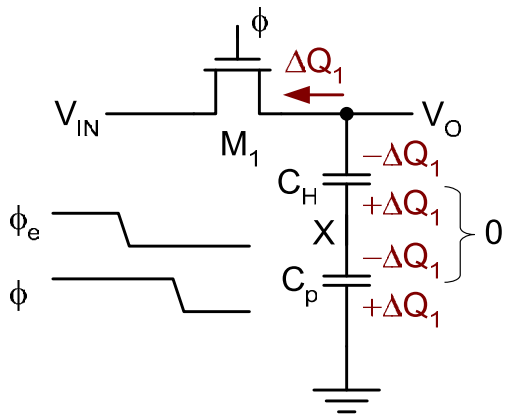
Bottom Plate Sampling (4)



$$\Delta Q_1 \cong \alpha_1 WLC_{ox}(\phi_H - V_{IN} - V_{tn})$$

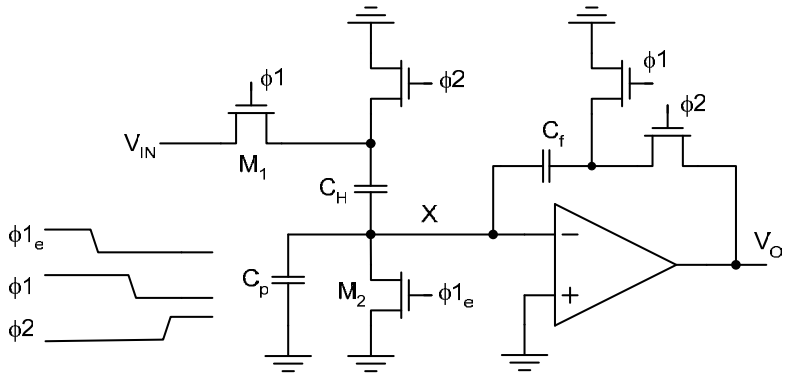
- No, of course not
  - There must be some parasitic cap, e.g. M_2 drain-to-bulk capacitance
- So, in real life, M1 does inject charge onto C_H
  - How much?
- Since M_1 sees C_H in series with C_p, α_1 and thus ΔQ_1 may be fairly small…
  - Not all that convincing…
- Fortunately, there's another trick we can pull

Bottom Plate Sampling (5)

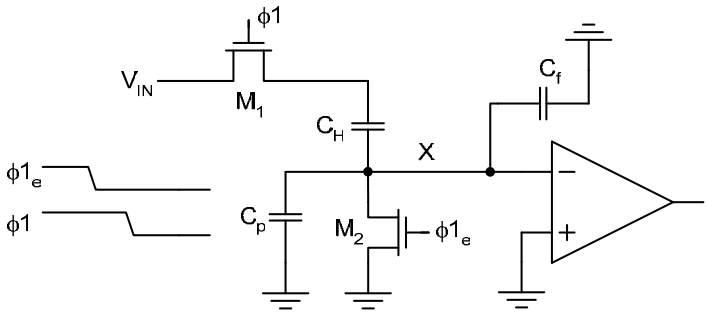


- Interesting observation
  - Even if M_1 injects some charge onto C_H, the total charge at node X cannot change!
- Idea
  - Process total charge at node X instead of looking at voltage across C_H

Charge Redistribution Track&Hold

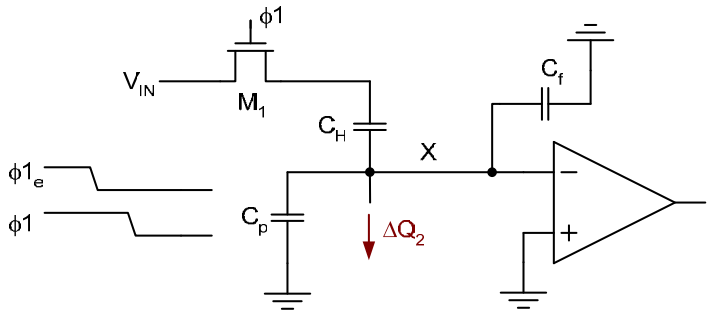


Circuit during ϕ1



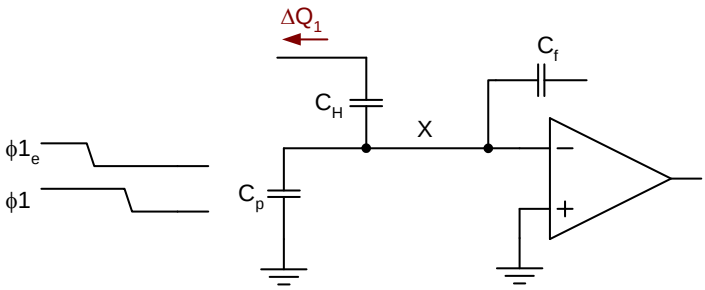
- Total charge at node X: $Q_{X0} = -C_H V_{IN}$

Circuit with ϕ1e Going Low



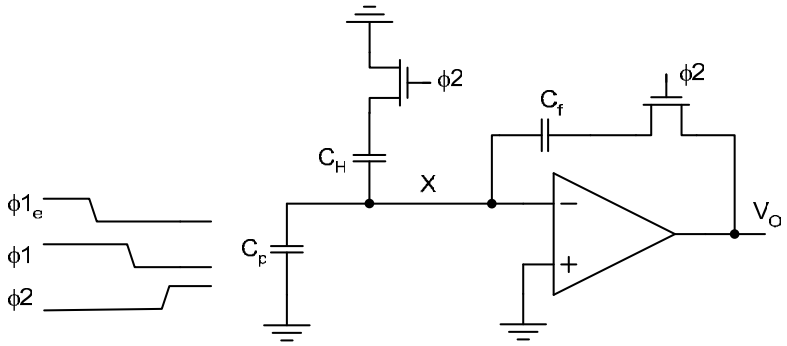
- Total charge at node X: $Q_{X1} = -C_H V_{IN} - \Delta Q_2$

Circuit with ϕ1 Going Low



- Charge injection ΔQ_1 leads to changes in voltage across all capacitors, but total charge at X remains unchanged!

Circuit During ϕ2



- OpAmp forces voltage at node X to zero
  - Means that charge at node X must redistribute among capacitors

Charge Conservation

Sampled Charge: $Q_{X1} = -C_H V_{IN} - \Delta Q_2$

After Redistribution: $Q_{X2} = -C_f V_O$

Charge Conservation:

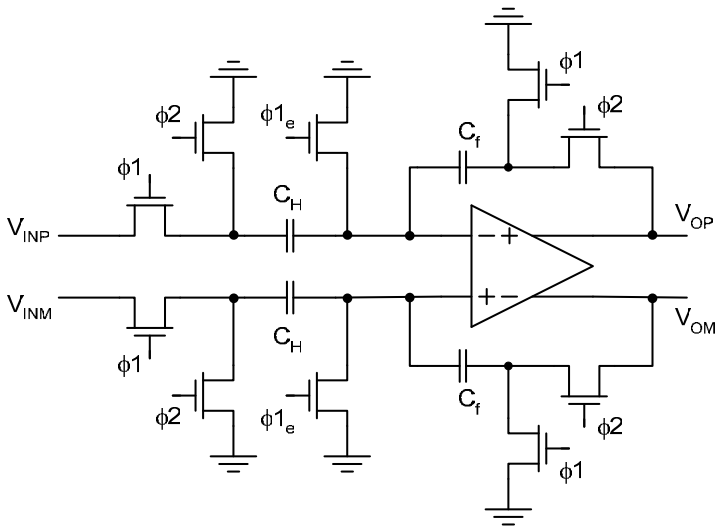
$$Q_{X1} = Q_{x2}$$

$$-C_H V_{IN} - \Delta Q_2 = -C_f V_O$$

$$\therefore V_O = \frac{C_H}{C_f} V_{IN} + \frac{\Delta Q_2}{C_f}$$

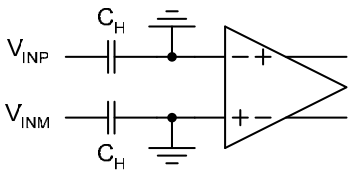
- Output has signal independent offset
  - Can easily cancel through fully differential implementation

Fully Differential Circuit

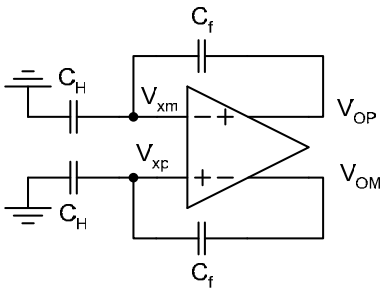


Analysis (1)

During ϕ1



During ϕ2



$$Q_{1m} = -C_H V_{INP} + \Delta Q \qquad Q_{2m} = C_H V_{xm} - C_f \left(V_{OP} - V_{xm}\right)$$

$$Q_{1p} = -C_H V_{INM} + \Delta Q \qquad Q_{2p} = C_H V_{xp} - C_f \left(V_{OM} - V_{xp}\right)$$

$$1)\quad Q_{1-} = Q_{2-} \qquad V_{xm} = V_{xp} \qquad \frac{V_{OP} + V_{OM}}{2} = V_{OC}$$

$$2)\quad Q_{1+} = Q_{2+}$$

Analysis (2)

- Subtracting 1) and 2) yields

$$V_{OP} - V_{OM} = \frac{C_s}{C_f}(V_{INP} - V_{INM})$$

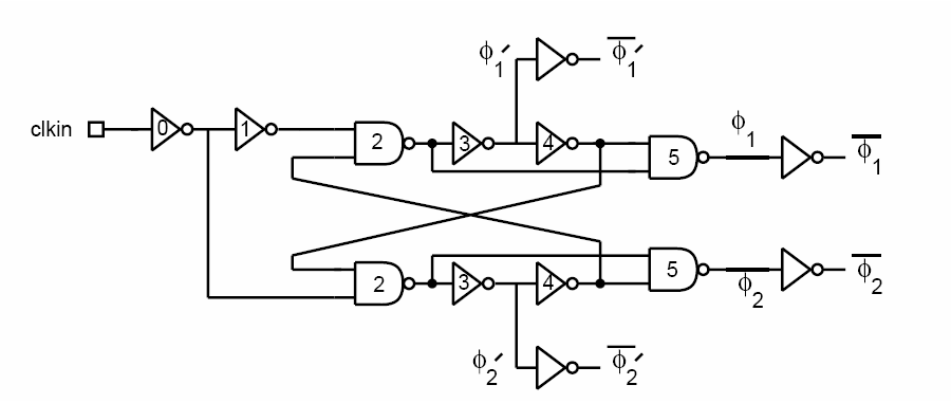
- Adding 1) and 2) yields

$$-C_H(V_{INP} + V_{INM}) + 2\Delta Q = (C_H + C_f)(V_{xp} + V_{xm}) - C_f(V_{OP} + V_{OM})$$

$$V_{xc} = \frac{\Delta Q}{C_H + C_f} + \frac{C_f}{C_H + C_f} V_{OC} - \frac{C_H}{C_H + C_f} V_{IC}$$

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

Clock Generation



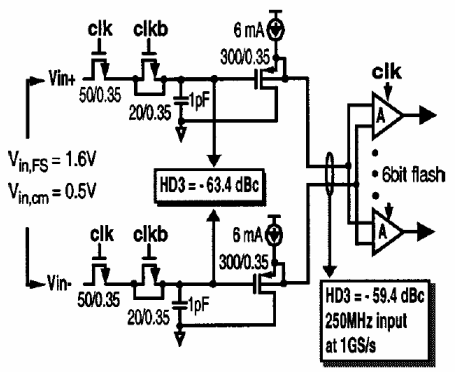
[A. Abo, "Design for Reliability of Low-voltage, Switched-capacitor Circuits," PhD Thesis, UC Berkeley, 1999]

More on T/H Circuits

- Implementation examples
  - Low precision, high-speed T/H
  - Charge redistribution T/H with common mode cancellation
  - Flip-around T/H
- Sampling network design considerations
- What limits the linearity of a bottom plate sampling circuit?

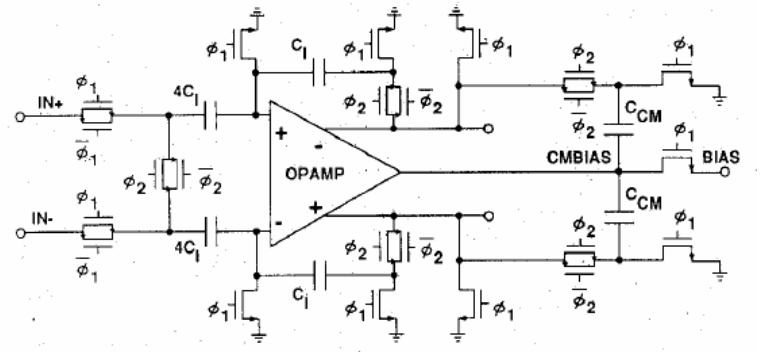
Low Precision, High-Speed T/H

- Important to note that for low resolution, high speed applications, a "simple" T/H circuit may suffice
  - No bottom plate sampling, no charge redistribution
  - Can use source follower to buffer sampled signal



[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]

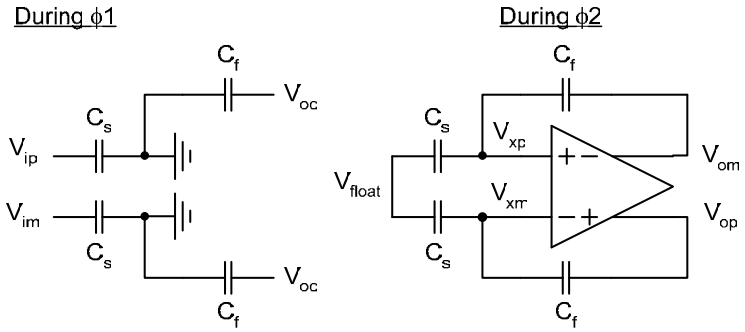
T/H with Common Mode Cancellation



[S.H. Lewis & P.R. Gray, "A Pipelined 5 MSample/s 9-bit Analog-to-Digital Converter", IEEE J. Solid-State Ckts, 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}

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

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

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

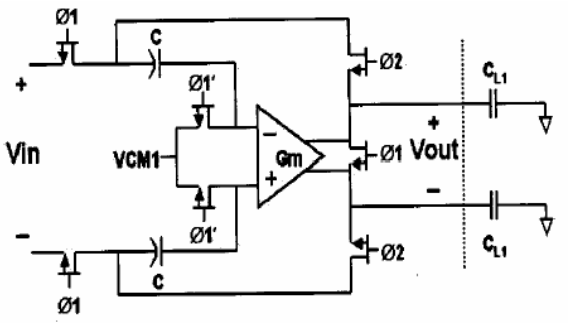
Analysis (2)

- Common mode charge conservation at amplifier inputs

$$-V_{ic}\cdot C_s - V_{oc}\cdot C_f = -\left(V_{float} - V_{xc}\right)\cdot C_s - \left(V_{oc} - V_{xc}\right)\cdot C_f$$
$$-V_{ic}\cdot C_s = -\left(\left[V_{ic} + V_{xc}\right] - V_{xc}\right)\cdot C_s + V_{xc}\cdot C_f$$
$$0 = V_{xc}$$

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

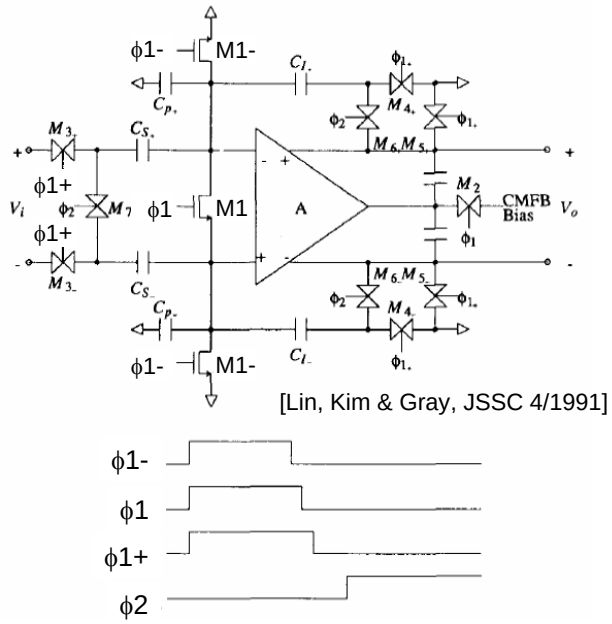
Flip-Around T/H



[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

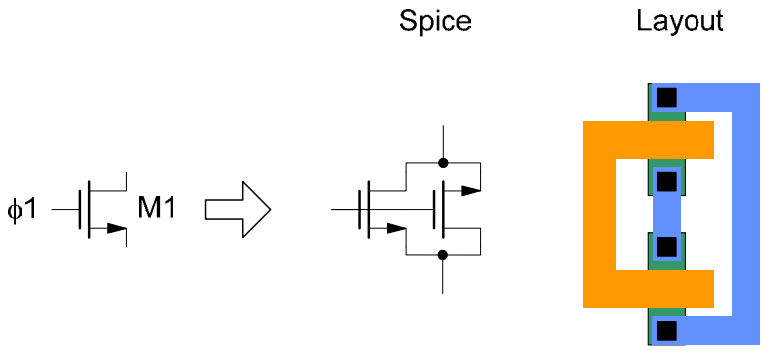
Sampling Network Design Considerations (1)



[Lin, Kim & 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, 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

Sampling Network Design Considerations (2)

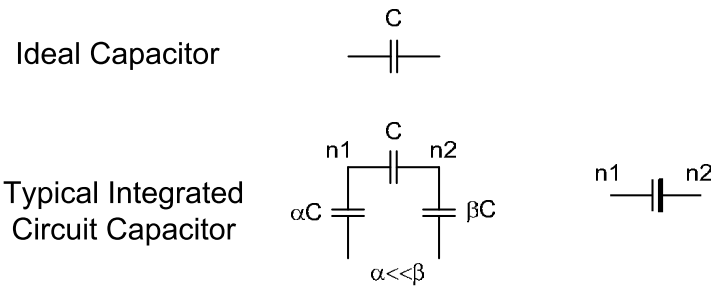


- Use antiparallel devices to implement M1
  - Needed in simulation to guarantee circuit symmetry
    - E.g. BSIM model is not necessarily perfectly symmetric with respect to drain/source!
  - Needed in layout to ensure symmetry in presence of drain/source asymmetry due to processing artifacts

Linearity Limitations

- Linearity of bottom plate sampling circuits is affected mainly by two effects
  - Track mode nonlinearity due to R=f(V_{in})
    - Can try to mitigate using clock bootstrapping and proper partitioning of total path resistance
    - Most detrimental at high frequencies
  - 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
    - Clock bootstrapping helps; ultimately limited by body effect
    - Often fairly independent of frequency (somewhat dependent on exact realization of top plate switch)
- In high speed designs, can achieved SFDR up to ~100dB at low input frequencies, ~80dB up to a few hundreds of MHz

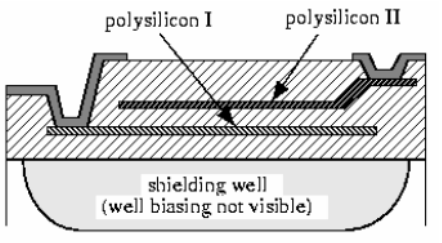
A Note on Integrated Capacitors



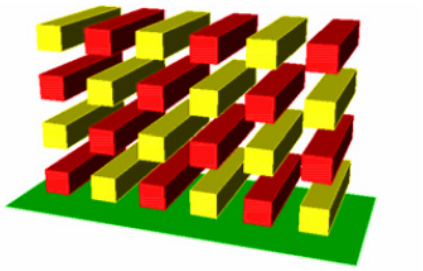
- Node n1 is usually the "physical" top plate of the capacitor
  - Makes nomenclature very confusing, since this plate is typically used as the "electrical" bottom plate in a sampling circuit (in the context of "bottom plate sampling")
- EE315 technology values
  - α=1%, β=10%

Various Capacitor Cross Sections

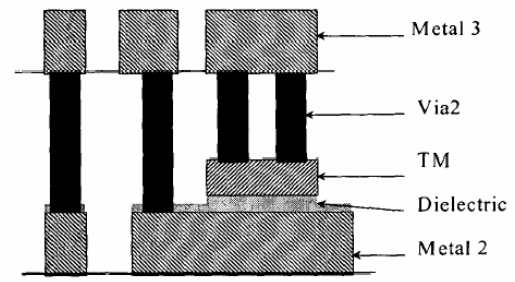
Poly-Poly Capacitor



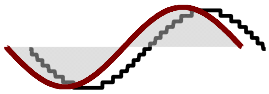
Metal-Metal Comb Capacitor



MIM Capacitor
(Metal-Insulator-Metal)



Lecture 8
Switched Capacitor Circuit Examples and Analysis

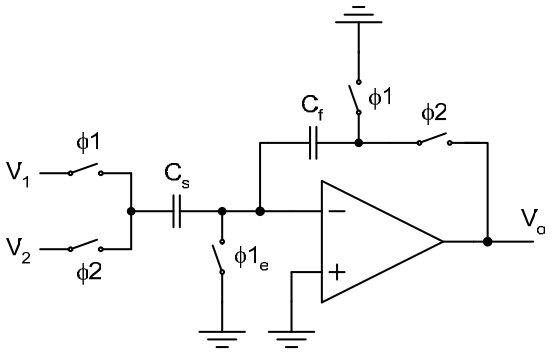


Boris Murmann
Stanford University
murmann@stanford.edu

Switched Capacitor Circuits

- The discussed T/H circuits are a subset of a much broader class of circuits for "discrete time," charge-based analog signal processing
- Other switched capacitor (SC) circuit examples
  - SC difference amplifiers
    - Used e.g. in pipeline ADCs
  - SC integrators
    - Used e.g. in sigma-delta ADCs
  - Passive charge redistribution networks
    - Used e.g. in DACs, successive approximation ADCs
  - SC biquads
    - Used to implement second order filter sections
  - …

SC Difference Amplifier

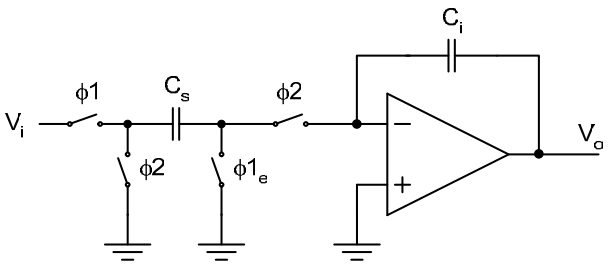


$$V_1 C_s = V_o C_f + V_2 C_s$$

$$V_o = \frac{C_s}{C_f}(V_1 - V_2)$$

- Useful for computing differences of signals
  - Application example: pipeline ADC (more later)

Integrator

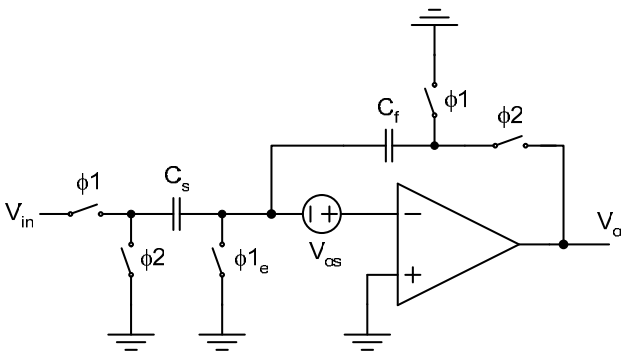


- C_i accumulates charge packets acquired during $\phi1$
  - "Discrete time integrator"
- Used e.g. in switched capacitor sigma-delta ADCs (more later)

Analysis of SC Circuit Nonidealities

- Amplifier offset
  - Several ways to compensate (if needed)
  - See e.g.
    - C.C. Enz & G.C. Temes, "Circuit techniques for reducing the effects of op-amp imperfections: Autozeroing, correlated double sampling, and chopper stabilization," Proc. IEEE, pp. 1584-1614, Nov. 1996.
- Finite bandwidth and slew rate in amplifier
- Nonzero switch time constant
  - Typically make switches about 5-10x faster than amplifier
- Electronic noise from switches and amplifier
- We'll look at these design aspects using a charge redistribution T/H circuit as an example

Offset

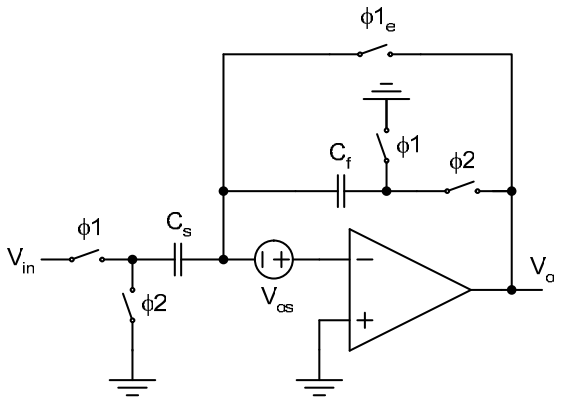


$$V_{in}C_s = (V_o + V_{os})C_f + V_{os}C_s$$

$$V_o = \frac{C_s}{C_f}V_{in} - \left(1 + \frac{C_s}{C_f}\right)V_{os}$$

- Amplified by (1+Gain)
- Often not a big issue

Auto-Zero Technique

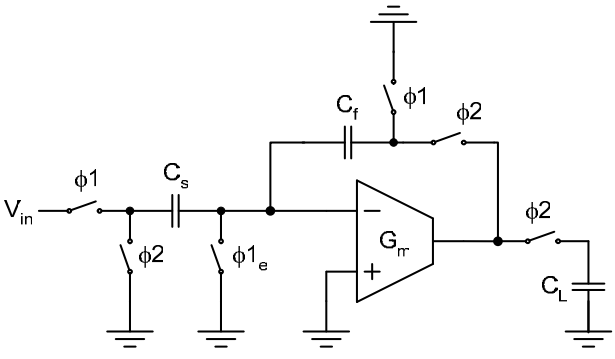


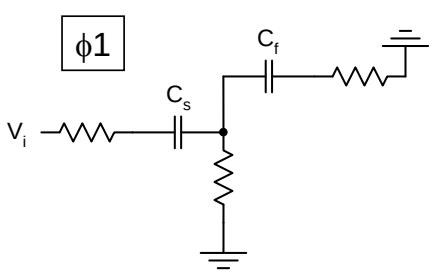
- Perfect cancellation, assuming infinite amplifier gain
  - Can show that finite gain limits achievable accuracy
- Additional caveats
  - In practice, offset tends to be dominated by mismatch in charge injection (fully differential circuit)
  - Amplifier must be unity gain stable!
    - May need to push nondominant poles to very high frequencies

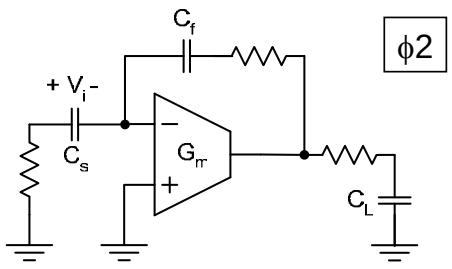
$$(V_{in}+V_{os})C_s+V_{os}C_f=(V_o+V_{os})C_f+V_{os}C_s$$

$$V_o=\frac{C_s}{C_f}V_{in}$$

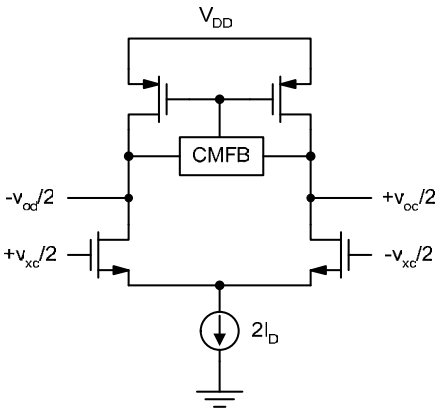
Settling & Noise Analysis

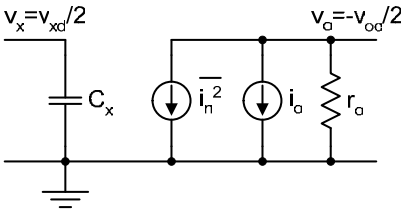






Basic OTA Model for Hand Analysis

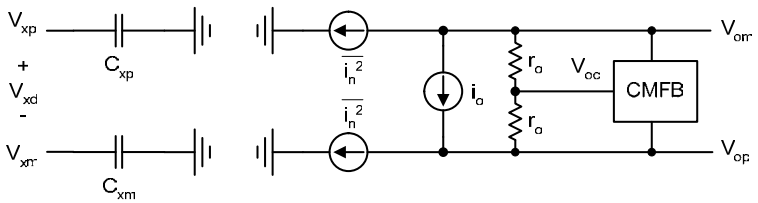




$$i_o = \begin{cases} g_m v_x & for \quad |g_m v_x| < I_D \\ I_D \cdot sign(v_i) & else \end{cases}$$

$$C_x = \frac{g_m}{2\pi f_T} \qquad r_o = \frac{a_{vo}}{g_m} \qquad I_D = \frac{g_m}{(g_m / I_D)} \qquad \frac{\overline{i_n^2}}{\Delta f} = n_f \frac{8}{3} kT \cdot g_m$$

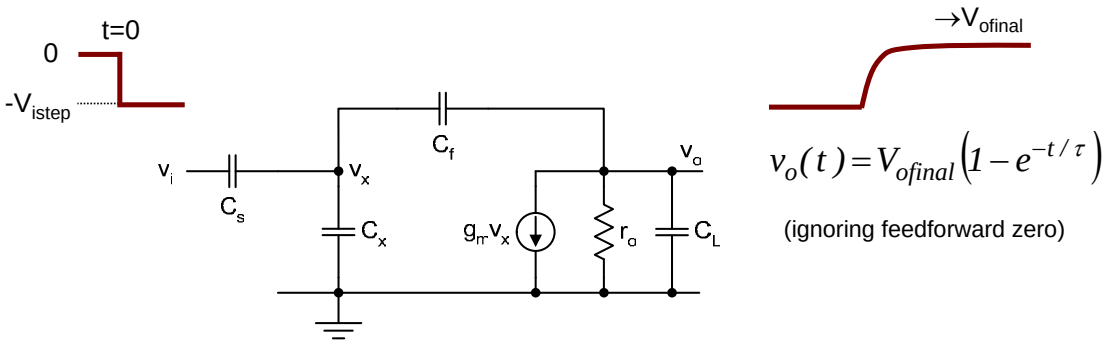
Simulation Model



$$C_{xp,m} = \frac{g_m}{2\pi f_T}$$

- HSpice model "ota1" (in ee315_hspice.txt)

Linear Settling



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

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

Static Settling Error

- Ideal output voltage for t→∞

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

- Detailed analysis shows
  - See e.g. EE214

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

- Static settling error

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

Dynamic Settling Error

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

$$N = \frac{t}{\tau} = -ln(\varepsilon_d)$$

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

Time Constant

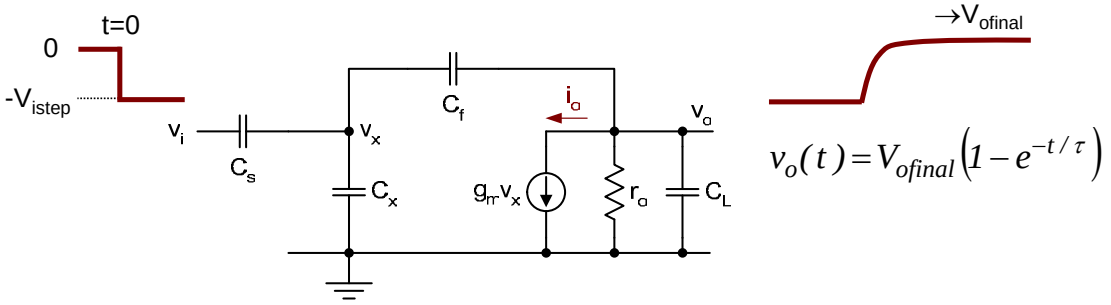
- Detailed analysis shows
  - See e.g. EE214

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

- Effective load capacitance is explicit load plus loading from feedback network

$$C_{Leff} = C_L + (1 - \beta) \cdot C_f$$

Transconductor Current



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

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

- Peak current occurs at t=0

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

Slewing

- The amplifier on slide 18 can deliver a maximum current of I_D
  - If $|i_o|_{max} > I_D$, slewing occurs

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

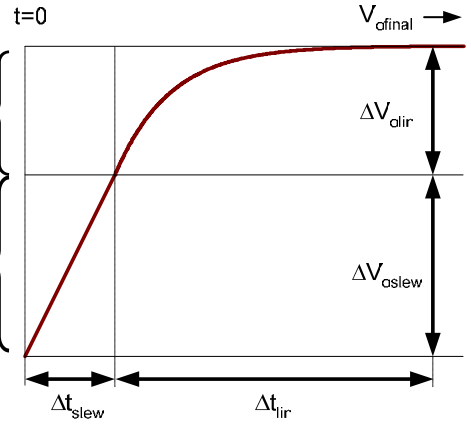
$$C_{Leff} \frac{V_{ofinal}}{\frac{1}{\beta} \cdot \frac{C_{Leff}}{g_m}} > I_D \qquad \Rightarrow \frac{g_m}{I_D} > \frac{1}{\beta V_{ofinal}}$$

- 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)
    - Output voltage swing is small

Output Waveform with Initial Slewing

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

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



- Continuous derivative in the transition slewing→linear requires

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

Dynamic Error with Slewing

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

- Note that these equations are valid for the half circuit
  - $\Delta V_{odslew}=2\Delta V_{oslew}$, $\Delta V_{odlin}=2\Delta V_{olin}$, $V_{odfinal}=2V_{ofinal}$
- Using the above result, we can now calculate the dynamic error during the final linear settling portion

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

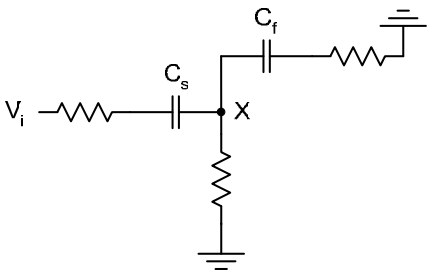
$$\varepsilon_{dynamic}(t) = \frac{v_o(t) - V_{final}}{V_{final}} = \frac{\Delta V_{oslew} + \Delta V_{olin}\left(1 - e^{-(t-\Delta t_{slew})/\tau}\right) - V_{ofinal}}{V_{ofinal}}$$

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

Noise Analysis

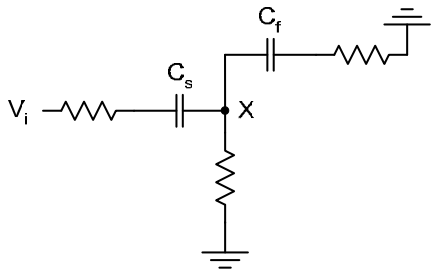
- Useful reference
  - Schreier et al., "Design-oriented estimation of thermal noise in switched-capacitor circuits," IEEE TCAS I, pp. 2358-2368, Nov. 2005.
- Switched capacitor circuits introduce noise in both clock phases
  - Tracking phase: kT/C noise from sampling switches
  - Redistribution phase: noise from switches and OTA
    - Switches tend to contribute much less noise than OTA
      - We'll take a closer look at that…
- If the noise in the two clock phases is uncorrelated, the total noise at the end of the redistribution phase can be found by superposition
  - Refer noise power of tracking phase to output and add to noise power introduced during redistribution

Tracking Phase (1)



- Variable of interest is total integrated "noise charge" at node X, $\overline{q_x^2}$
- 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

Tracking Phase (2)



- 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

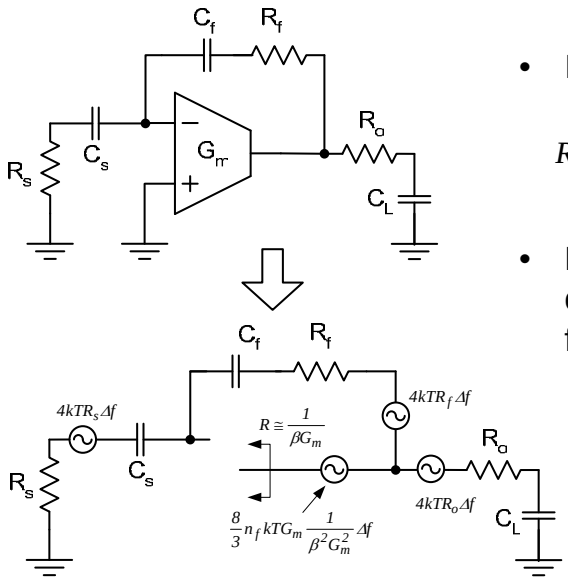
$$\overline{\frac{1}{2}\frac{q_x^2}{C_s + C_f}} = \frac{1}{2}kT$$

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

- Refer to output

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

Redistribution Phase (1)

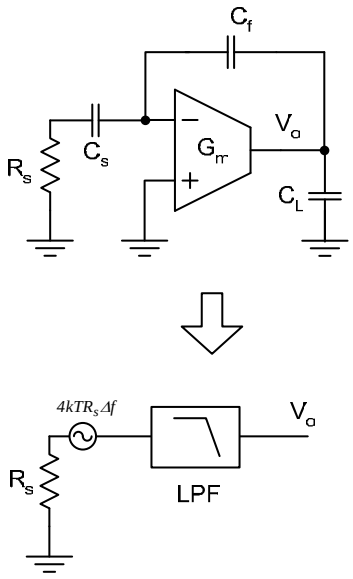


- In a proper design

$$R_f << \frac{1}{\beta^2 G_m}, \qquad R_o << \frac{1}{\beta^2 G_m}$$

- Hence, we can neglect noise contributions from R_f and R_o in first order noise calculations
  - Can always simulate to get more precise numbers…

Redistribution Phase (2)



- If LPF corner frequency is such that the effective "noise bandwidth" is

$$\Delta f = \frac{1}{4}\frac{1}{R_s C_{Leff}}$$

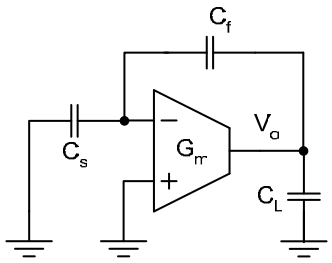
- Then $$\overline{v_{o,R_s}^2} = \frac{kT}{C_{Leff}}$$

- But, in this circuit

$$\Delta f = \frac{1}{4}\frac{\beta G_m}{C_{Leff}}$$

- So $$\overline{v_{o,R_s}^2} = \frac{kT}{C_{Leff}}\frac{R_s}{1/\beta G_m} << \frac{kT}{C_{Leff}}$$

Redistribution Phase (3)



- As we know from EE214, the total noise due to the single stage OTA is

$$\overline{v_{o,G_m}^2} = \frac{2}{3} n_f \frac{1}{\beta}\frac{kT}{C_{Leff}}$$

- This term is much larger than all other noise sources that we have considered in the redistribution phase, hence

$$\overline{v_{o,2}^2} \cong \frac{2}{3} n_f \frac{1}{\beta}\frac{kT}{C_{Leff}}$$

Total Noise

- Adding the noise contributions from tracking and redistribution phase we get

$$\overline{v_{o,tot}^2} = \overline{v_{o,1}^2} + \overline{v_{o,2}^2} \cong \frac{kT}{C_f}\left(1+\frac{C_s}{C_f}\right)+\frac{2}{3}n_f\,\frac{1}{\beta}\frac{kT}{C_{Leff}}$$

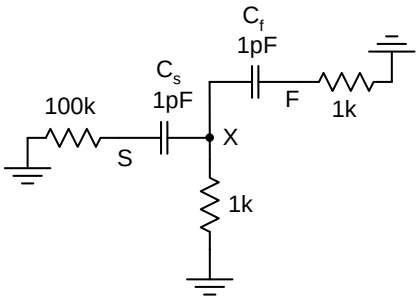
- If the circuit is fully differential, the above total noise power must be multiplied by two
  - Assuming that the noise in the two half circuits is uncorrelated
    - Usually the case, but beware of exceptions…

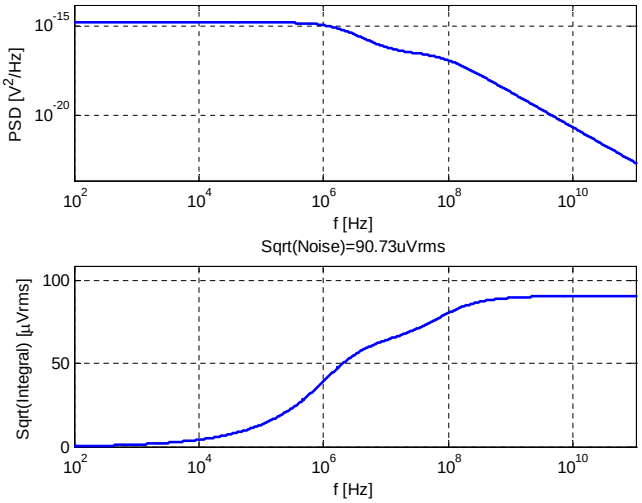
Noise Simulations

- Two ways to to simulate noise in switched capacitor circuits
- Basic Spice simulation using .noise
  - Must simulate noise in each clock phase separately
    - Activate ϕ1 switches, run .noise and integrate noise charge at relevant node over all frequencies
      - Refer to output to get output referred contribution
    - Activate ϕ2 switches, run .noise and integrate total noise at output
- Advanced simulators
  - E.g. SpectreRF, "periodic noise analysis"
  - Allows to simulate noise while switched capacitor circuit is clocked between ϕ1 and ϕ2
    - Noise from all phases is automatically added, all correlation taken care of
  - Good reference
    - K. Kundert, "Simulating switched-capacitor filters with SpectreRF," available at http://www.designers-guide.org/Analysis/sc-filters.pdf.

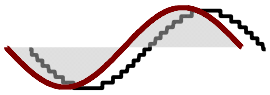
HSpice Example (Track Mode Noise Charge)

```
...
en vno 0  vcvs vol=( cs*v(x,s) + cf*v(x,f) )/cf'
.ac dec 100 100 100Gig
.noise v(vno) vdummy
```



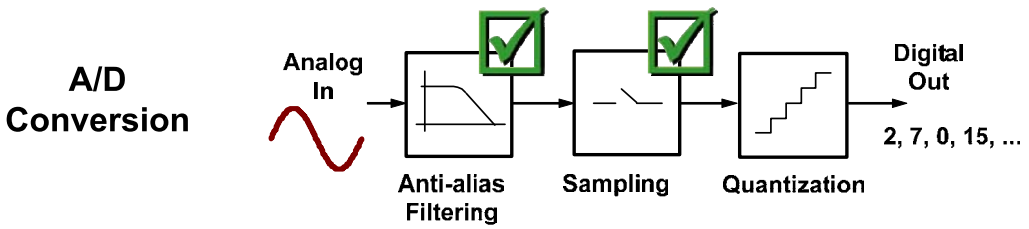


Lecture 9
Voltage Comparators



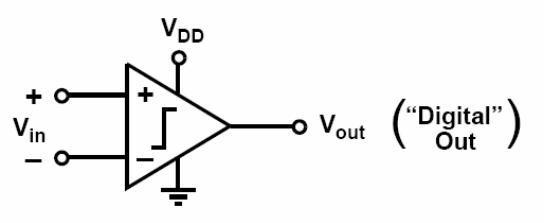
Boris Murmann
Stanford University
murmann@stanford.edu

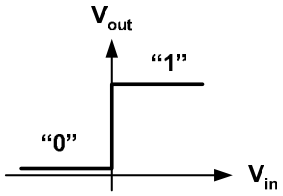
Recap



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

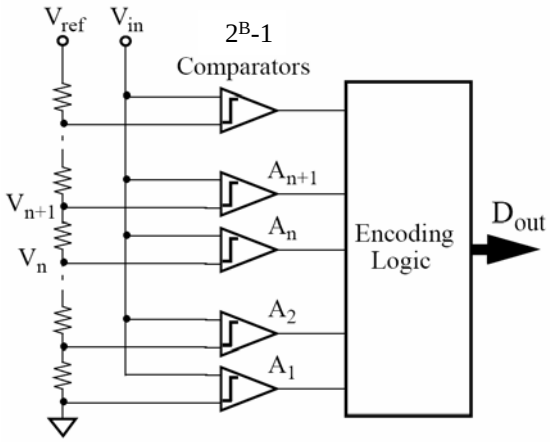
Ideal Voltage Comparator

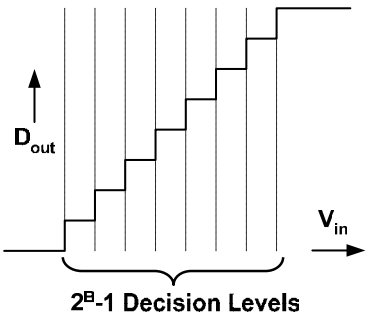




- Function
  - Compare the instantaneous values of two analog voltages (e.g. an input signal and a reference voltage) and generate a digital 1 or 0 indicating the polarity of that difference

Preview - Flash ADC

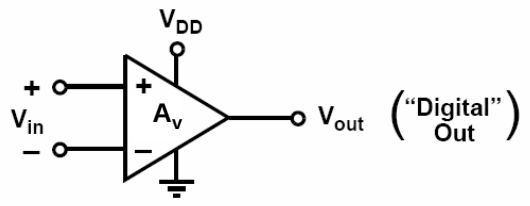


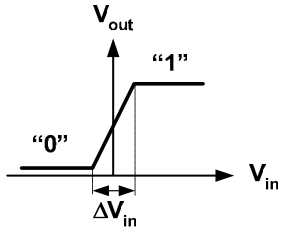


Design Considerations

- 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





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

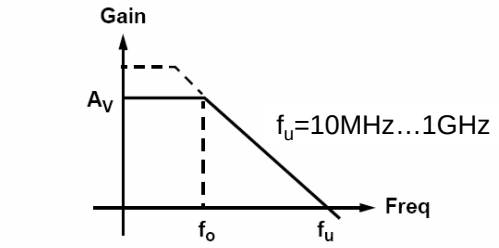
- E.g. 12-bit ADC, V_{DD}=1.8V, FSR=0.9V, $\Rightarrow$LSB=0.9V/4096
- For 1/2 LSB precision, we need

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

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?

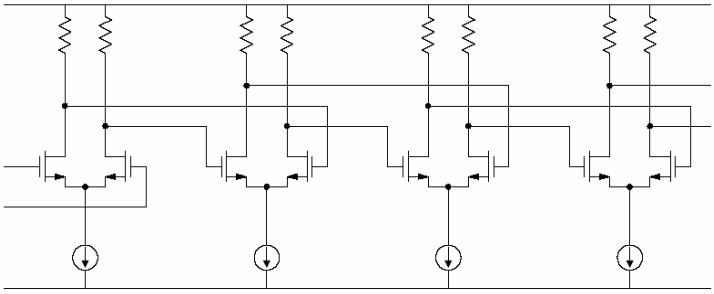


f_u = unity gain frequency, f_o = – 3dB frequency

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

- Way too slow!

Cascade of Open-Loop Amplifiers



In each stage: $\omega_u = \frac{g_m}{C_{gs}} \cong const.$ $A_0 = g_m R_L = \frac{\omega_u}{\omega_0}$ $\omega_0 = \frac{1}{RC} = \frac{\omega_u}{A_0}$

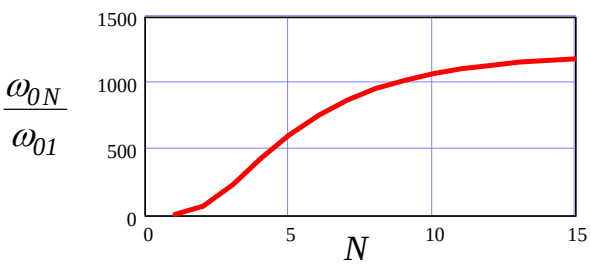
- Possible choices for a given, constant overall gain objective
  - Lots of stages with low gain
  - Only a few stages with moderate gain

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)

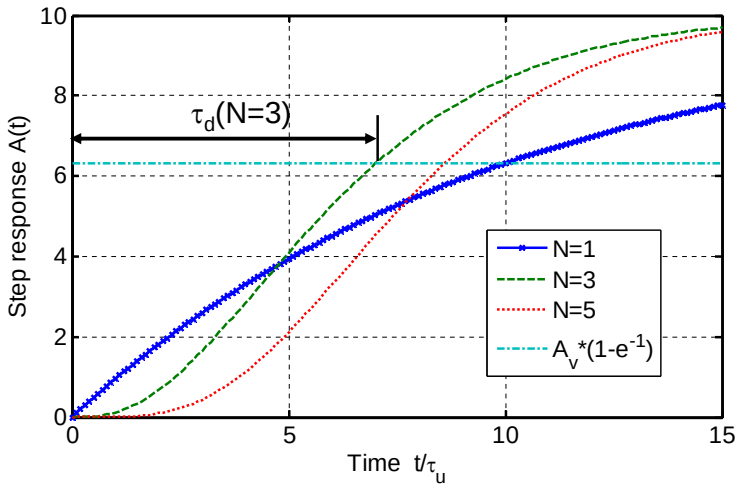
Step Response (1)

- In many cases (e.g. sampled data applications), it is more important to minimize the delay in response to an input step

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

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

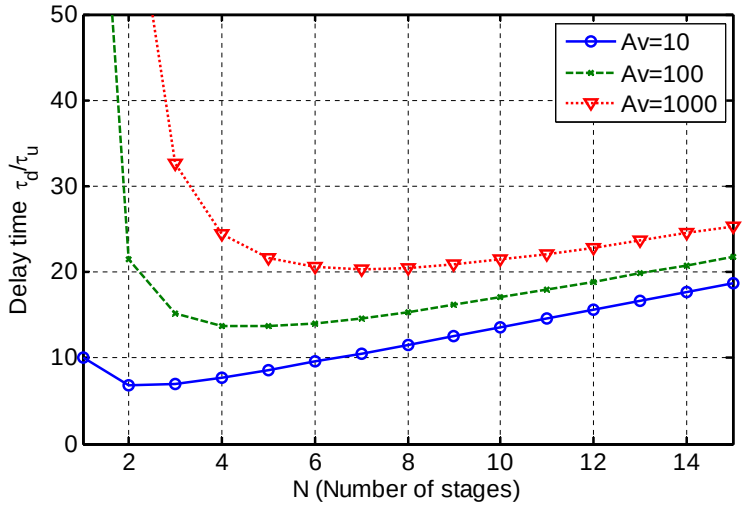
Step Response (2)



- Three stage amplifier wins! (for A_v=10)

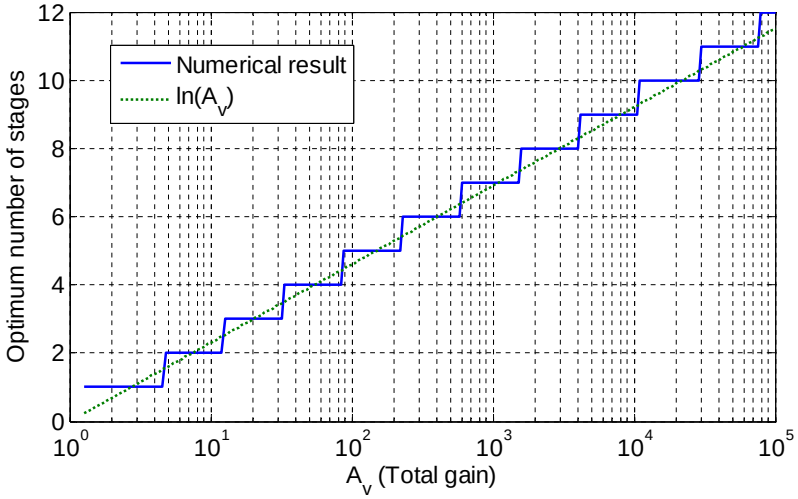
Delay versus Number of Stages

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



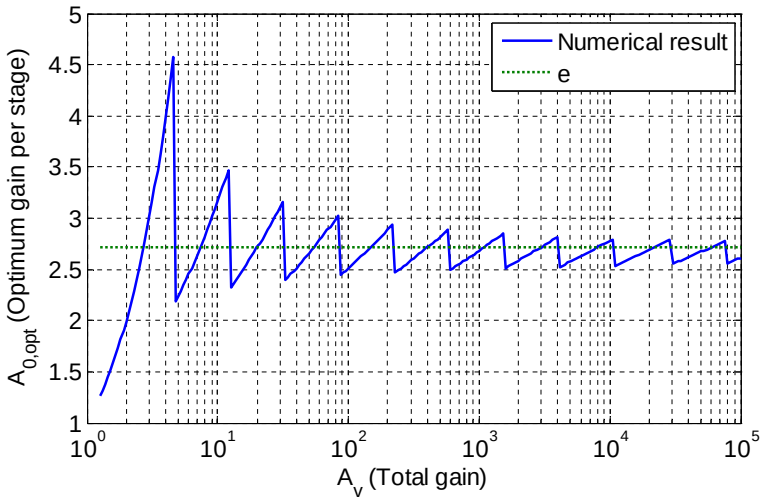
- Shallow minima!

Optimum Number of Stages



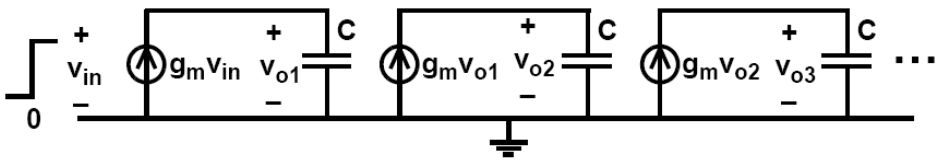
$$N_{opt} \cong ln(A_v)$$

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)

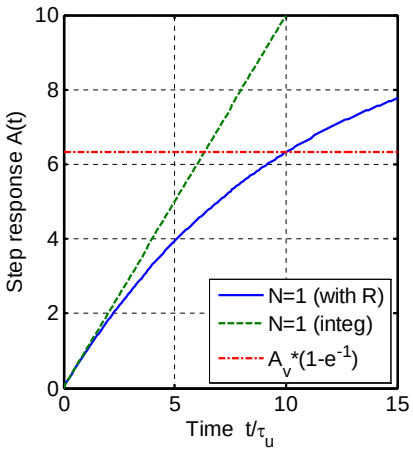


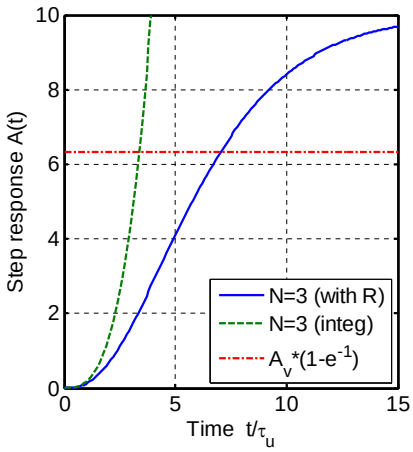
- Intuition
  - Load resistors (slide 9) shunt current away from load capacitance; this slows down amplification
  - Drop assumption $A_v=A_0{}^N$ to see what happens…
- Analysis

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

Cascade of "Integrators" (2)

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





Cascade of "Integrators" (3)

- 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

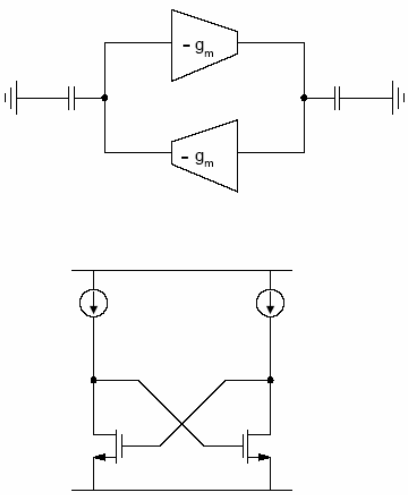
$$N_{opt} = 1.1\ln\left[A(\tau_d)\right] + 0.79 \qquad \text{[Wu, JSSC 12/1988)}$$

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

Regenerative Sense Amplifier (Latch)

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

$t \geq 0$ *enable positive feedback*



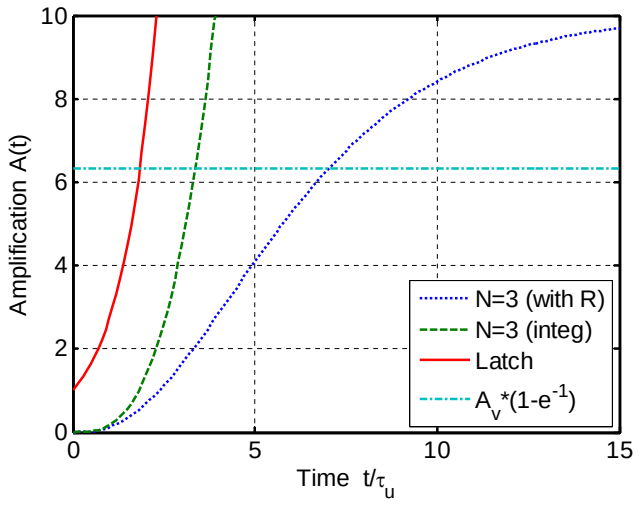
$$\frac{dv_1}{dt} = \frac{i_1(t)}{C} = \frac{-g_m v_2(t)}{C} \qquad \tau_u = \frac{C}{g_m}$$

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

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

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

Comparison

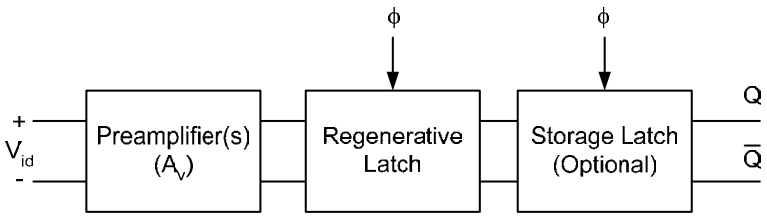


- Latch is much faster than cascade of amplifiers/integrators

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

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

Metastability (2)

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

- Assuming a uniform input signal distribution over some range

$$P\left(Error\right) = \frac{V_{id0\min}}{V_{id0\max}}$$

$$P(Error) = \frac{1}{A_v}\frac{V_{DD}}{V_{id0\max}}e^{-T_{\max}/\tau_u}$$

- For a B-bit Flash ADC

$$P(Error) = \frac{1}{A_v}\frac{V_{DD}}{\frac{V_{FS}}{2^B-1}}e^{-T_{\max}/\tau_u}$$

Metastability (3)

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

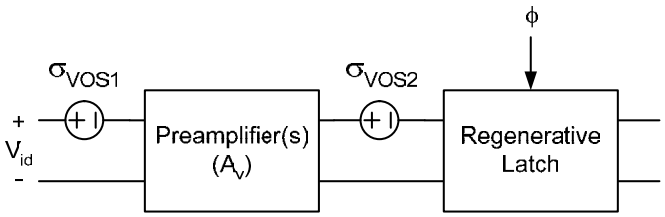
$$P(Error)=\frac{2}{3}\left(2^6-1\right)\cdot e^{-1000/32}\cong 10^{-12}$$

- Mean time to failure (MTF)

$$MTF=\frac{1}{P(Error)\cdot f_s}=\frac{1}{10^{-12}\cdot 0.5\cdot 10^9}s=2000s\cong 33\text{minutes}$$

- 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

Input Referred Offset



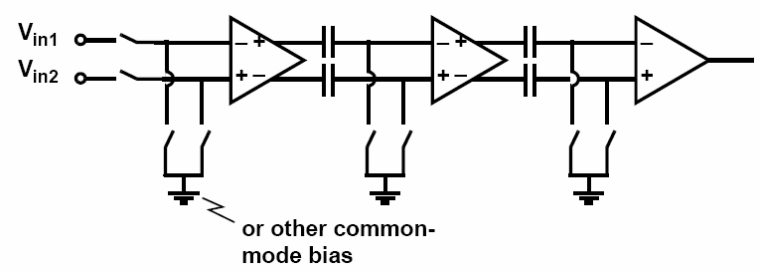
$$\sigma_{VOS}^2=\sigma_{VOS1}^2+\frac{1}{A_v^2}\sigma_{VOS2}^2$$

- Example: $\sigma_{VOS1}=3mV$, $\sigma_{VOS2}=30mV$, $A_v=10$

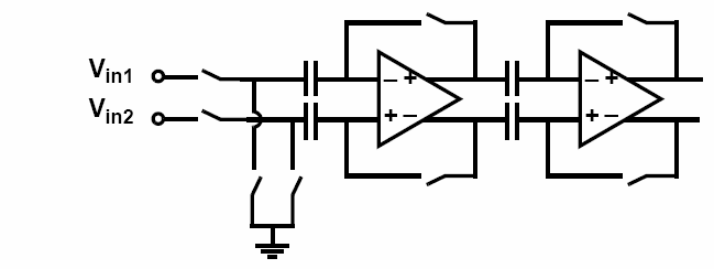
$$\sigma_{VOS}=\sqrt{(3mV)^2+\frac{1}{10^2}(30mV)^2}=4.2mV$$

Offset Cancellation

OUTPUT SERIES CANCELLATION

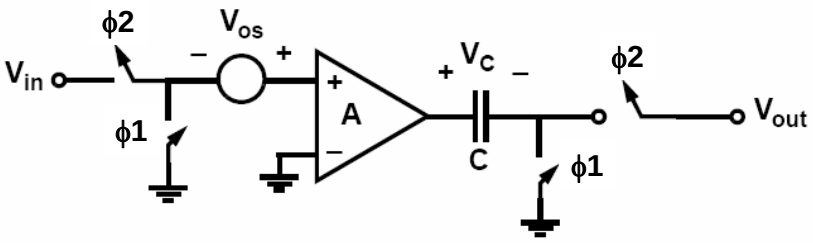


INPUT SERIES CANCELLATION



Output Series Cancellation

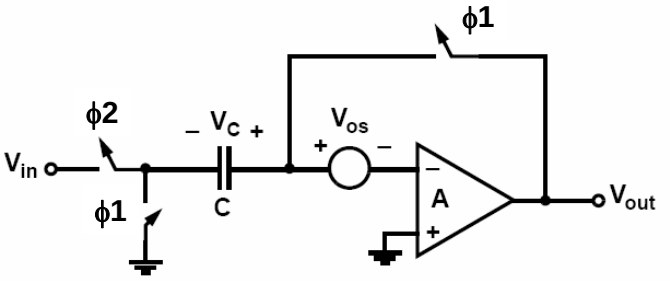
Ref: Poujois, et al., JSSC 8/78



- 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

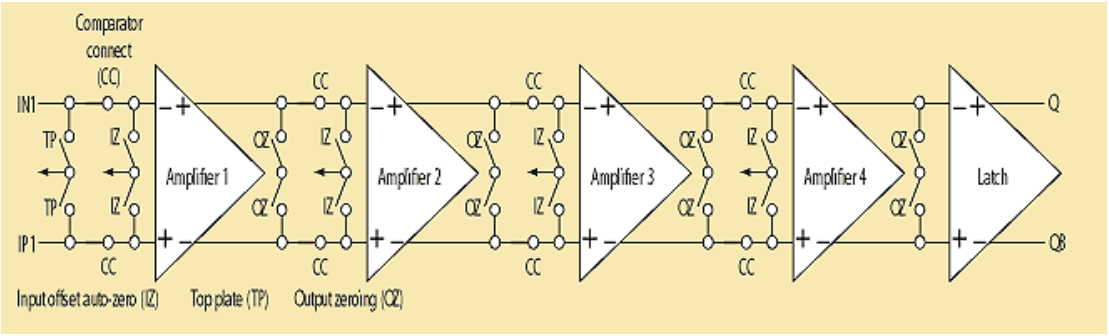
Refs: McCreary & Gray
Yee, et al.



- 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

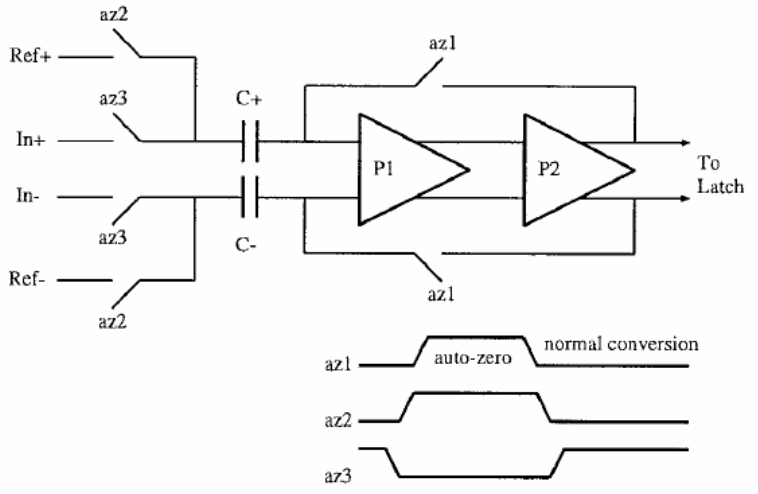
[http://www.elecdesign.com/Articles/Index.cfm?ArticleID=3956]



- 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)

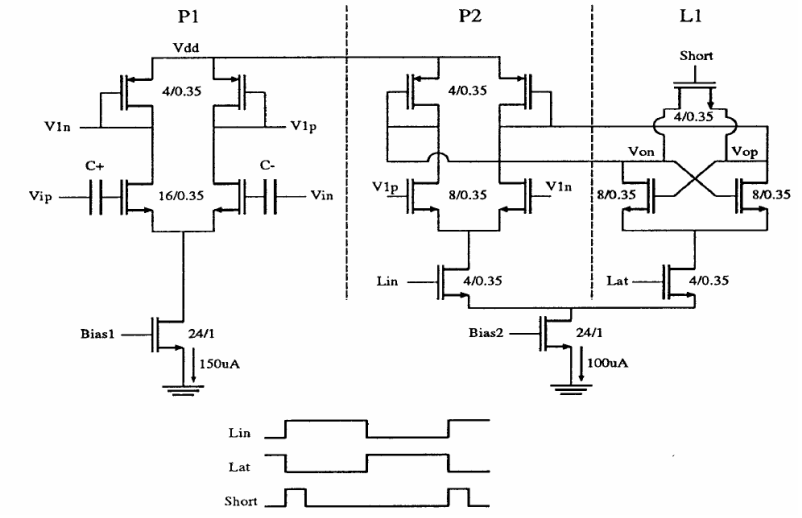
Comparator Examples (1)

- Mehr & Dalton, JSSC 7/1999



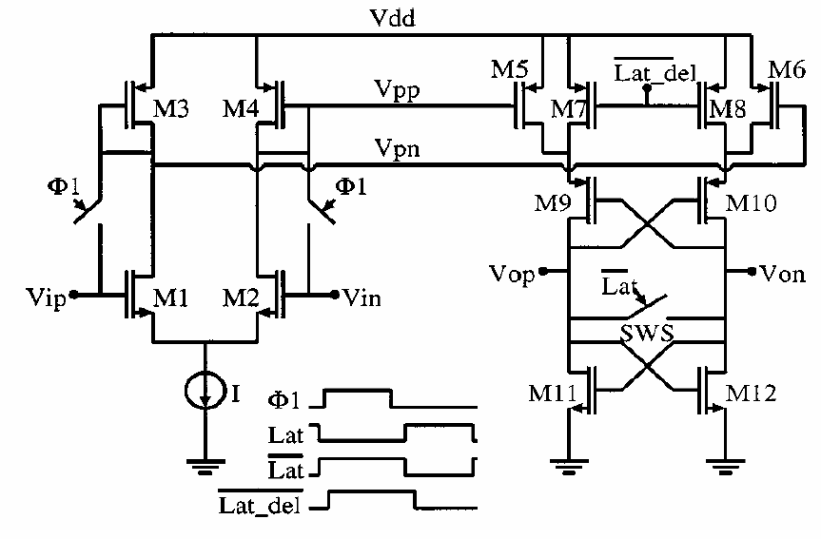
Comparator Examples (2)

- Mehr & Dalton, JSSC 7/1999



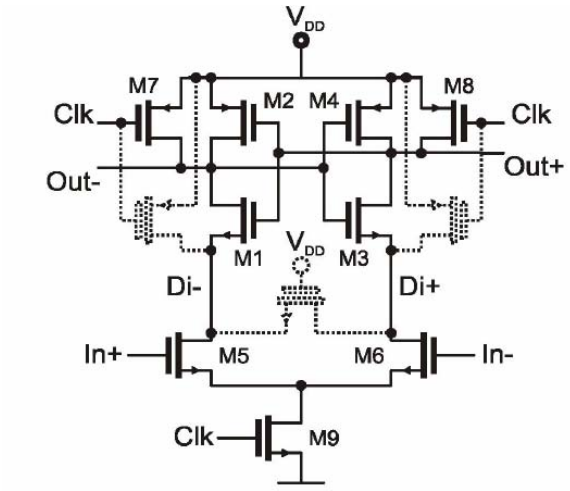
Comparator Examples (3)

- Mehr & Singer, JSSC 3/2000



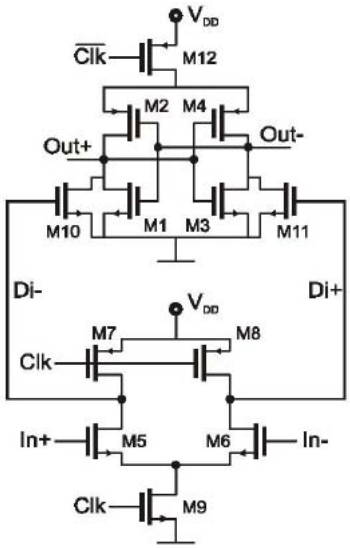
Comparator Examples (4)

- Purely dynamic "sense amplifier"
  - No DC current



Comparator Examples (5)

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



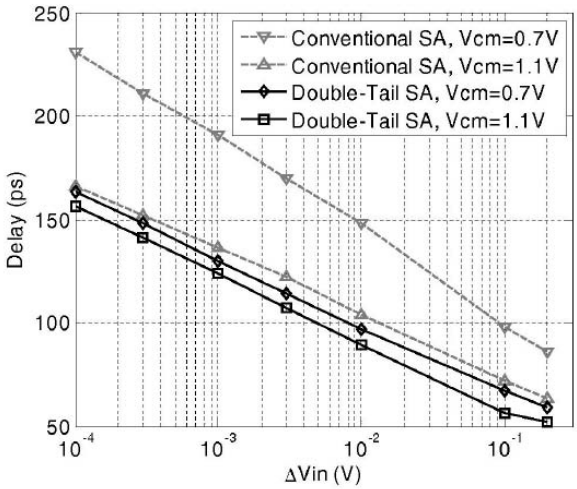


Figure 17.7.3: Simulated sense amplifier delays versus differential input voltage. The delay is the time between the clock edge and the instant when ΔOut crosses 1/2 V_{DD}.

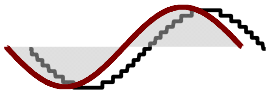
Selected References (1)

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.

Selected References (2)

9. 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.
10. 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.
11. 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.
12. 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.
13. 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.
14. 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.
15. A. Graupner, "A Methodology for the Offset-Simulation of Comparators," http://www.designers-guide.org/Analysis/comparator.pdf.
16. 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.
17. P.P. Nuzzo, et al., "Noise Analysis of Regenerative Comparators for Reconfigurable ADC Architectures, to appear, IEEE Trans. Circuits Syst. I, 2008.

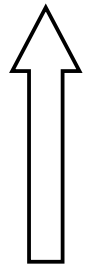
Lecture 10
Nyquist ADC Architectures
Flash ADCs



Boris Murmann
Stanford University
murmann@stanford.edu

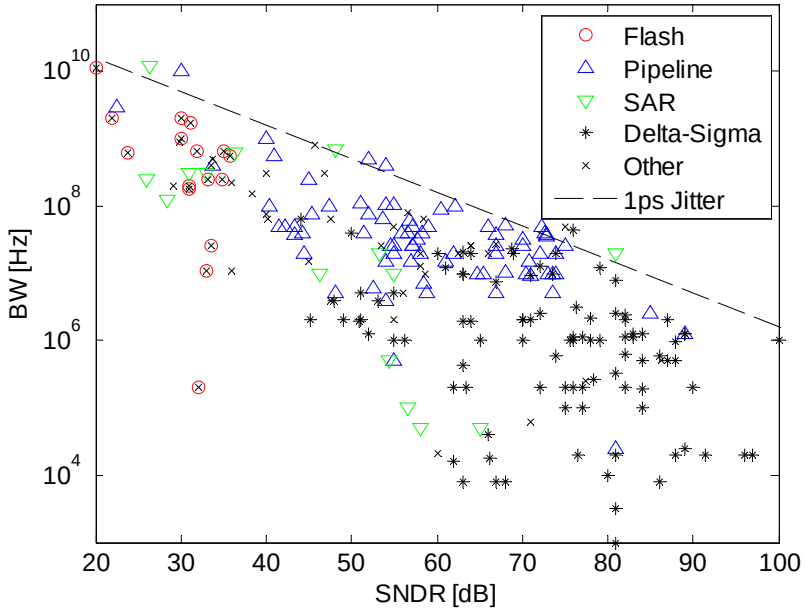
Nyquist ADC Architectures

- Nyquist rate
  - Word-at-a-time
    - E.g. 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

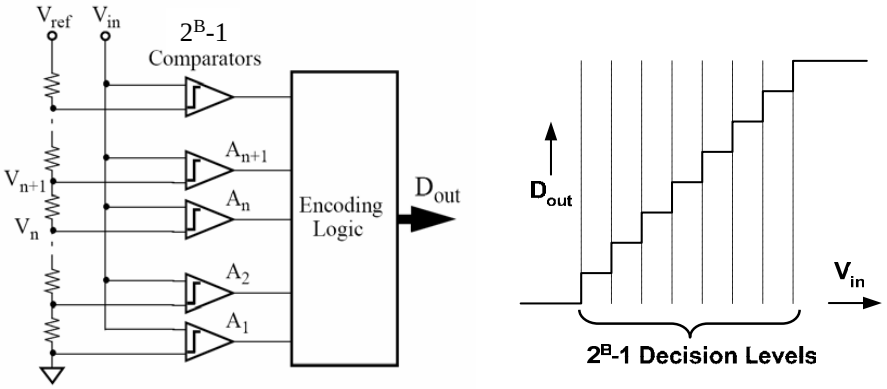


ADC Performance Survey (ISSCC & VLSI 97-08)

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



Flash ADC

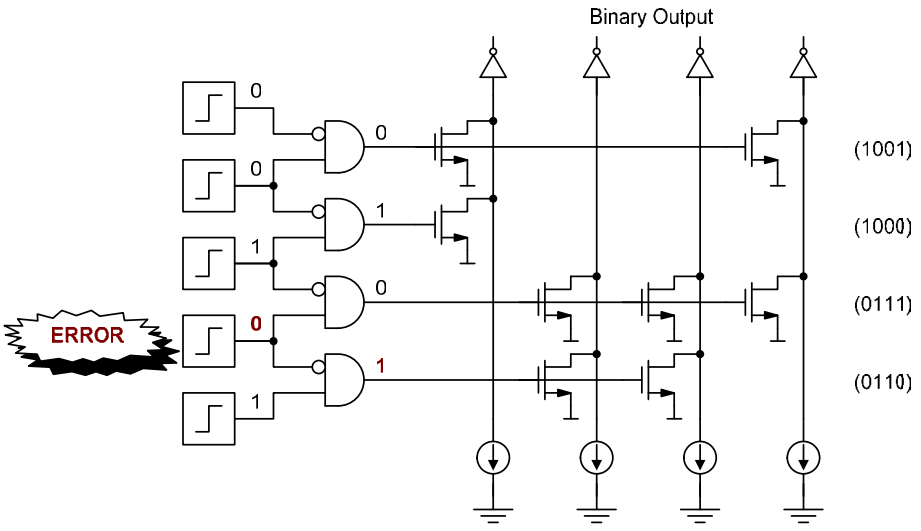


- Fast
  - Speed limited by single comparator plus encoding logic
- High complexity (2^B-1 comparators), high input capacitance
  - Typically use for resolution up to 6 bits

Limiting Error Sources

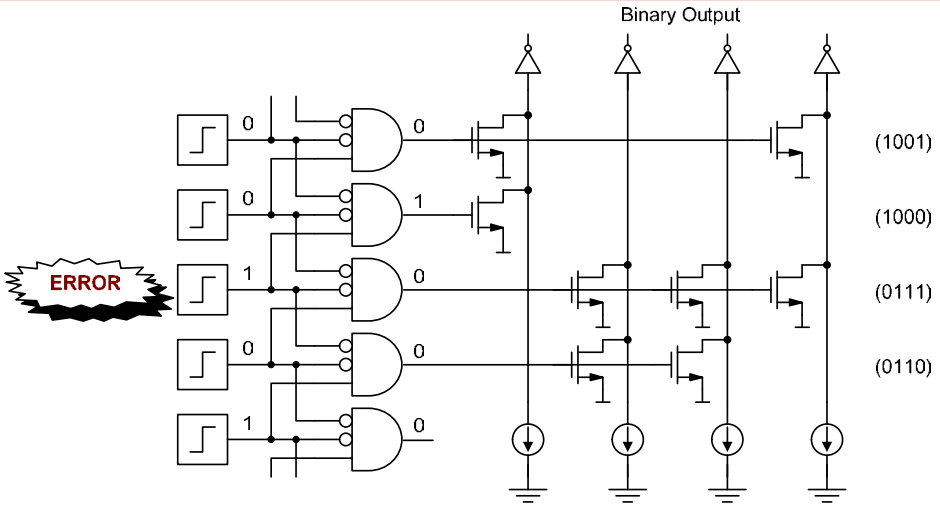
- Comparator input
  - Offset
  - Nonlinear input capacitance
  - Kickback noise (disturbs reference)
- Comparator output
  - Sparkle codes (… 111101000 …)
  - Metastability
    - Analog Devices application note: "Find Those Elusive ADC Sparkle Codes and Metastable States" http://www.analog.com/en/content/0,2886,760%255F788%255F91218,00.html
- Clock distribution and timing
  - Clock wiring can introduce significant delay
  - Comparators may sample signals at slightly different points due to mismatch or signal dependent sampling instant

Sparkle Codes



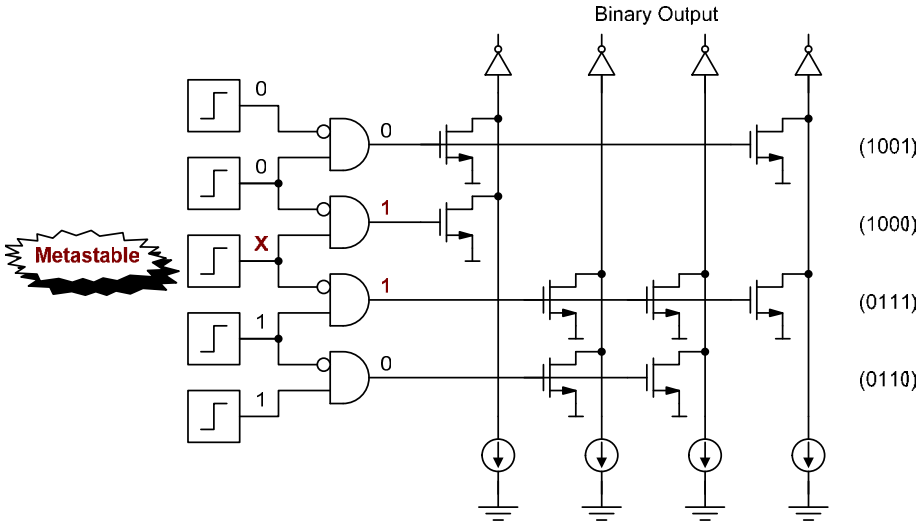
- Correct output: 1000, actual output: 1110 (!)

Sparkle Tolerant Encoder



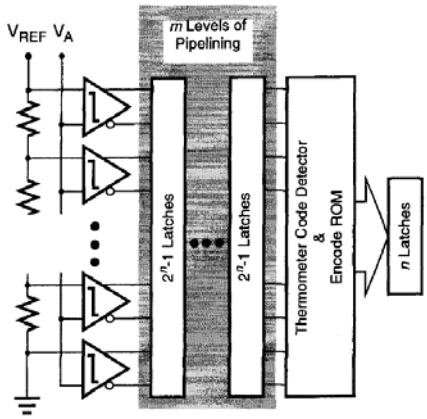
- Protects against isolated, single "bubbles"
- Reference: C. Mangelsdorf et al., “A 400-MHz Flash Converter with Error Correction,” IEEE J. Solid-State Ckts., pp. 997-1002, Feb. 1990.

Metastability



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

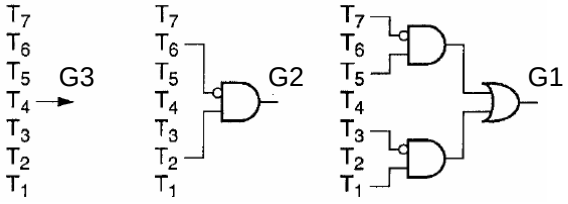
Solution 1: Latch Pipelining



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

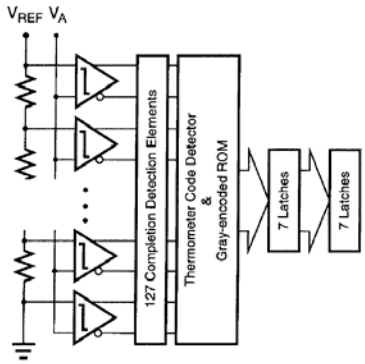
Solution 2: Gray Encoding

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

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

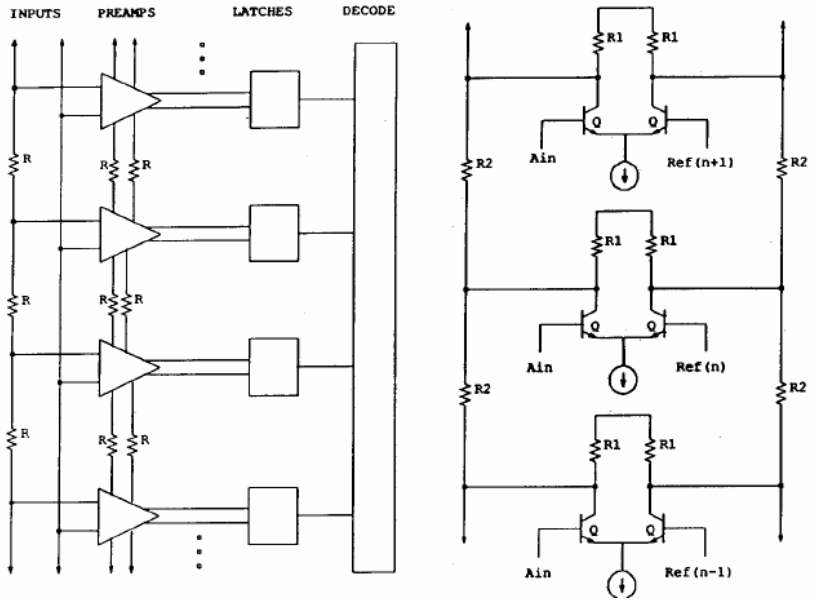
$$WL > \left(\frac{3A_{VT}}{2.8mV}\right)^2 = \left(\frac{3\cdot 4mV\mu m}{2.8mV}\right)^2 = 18.4\mu m^2 \Rightarrow W > \frac{18.4\mu m^2}{0.18\mu m} = 102\mu m$$

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 and/or postprocessing techniques

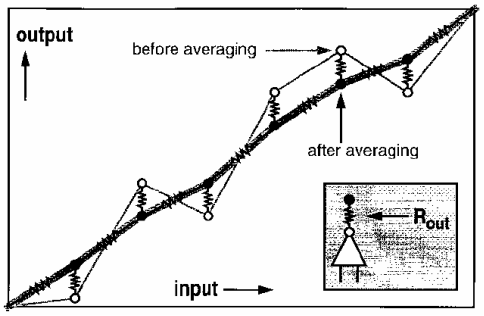
Offset Averaging (1)



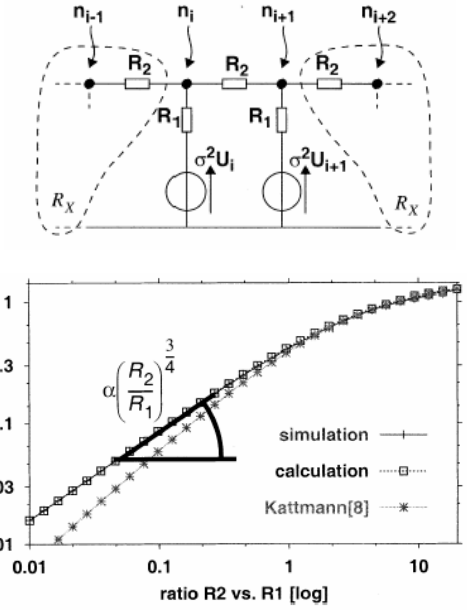
R2/R1=1.3

[Kattmann & Barrow, ISSCC 1991]

Offset Averaging (2)

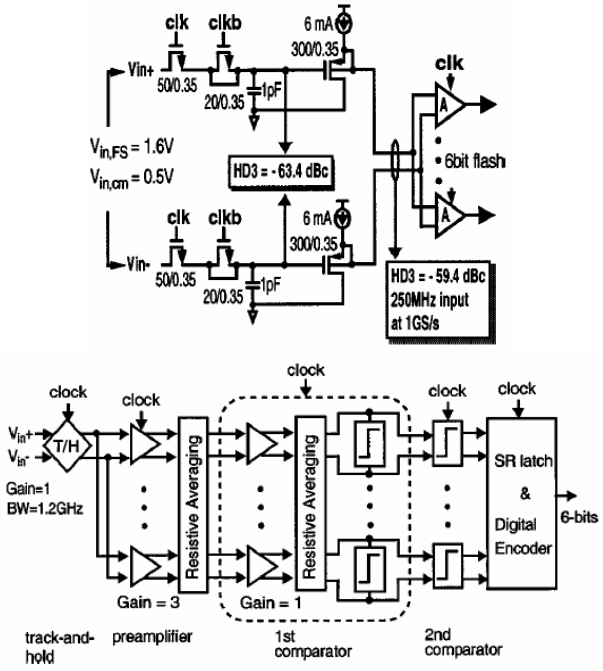


[Bult & Buchwald, JSSC 12/1997]



[Scholtens & Vertregt, JSSC 12/2002]

6-bit Flash ADC with Averaging

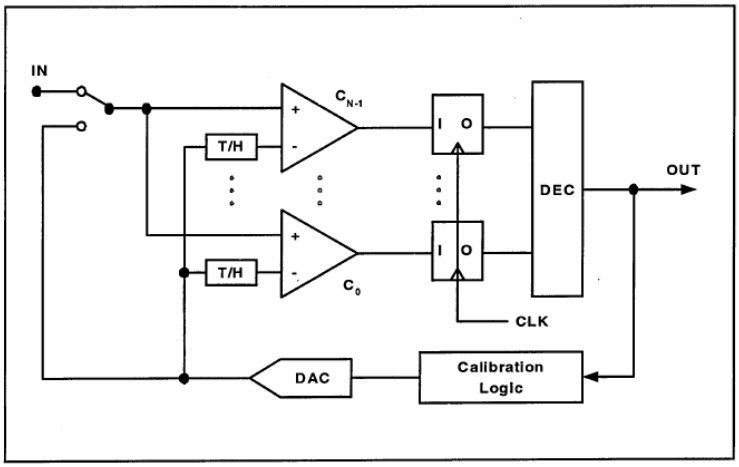


[Choi & Abidi, JSSC 12/2001]

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

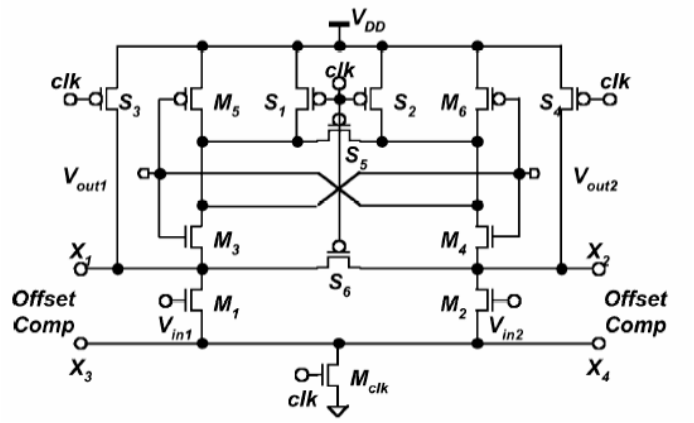
Averaging networks designed to reduce input referred offset by 3x

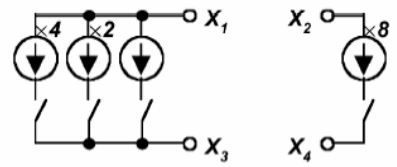
Offset Calibration



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.

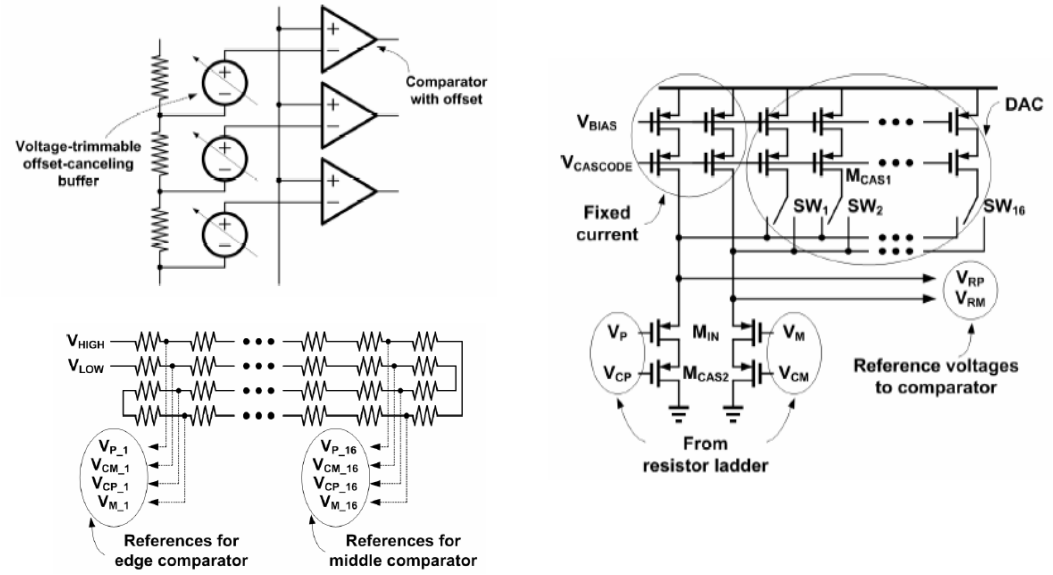
Comparator with Integrated Offset DAC





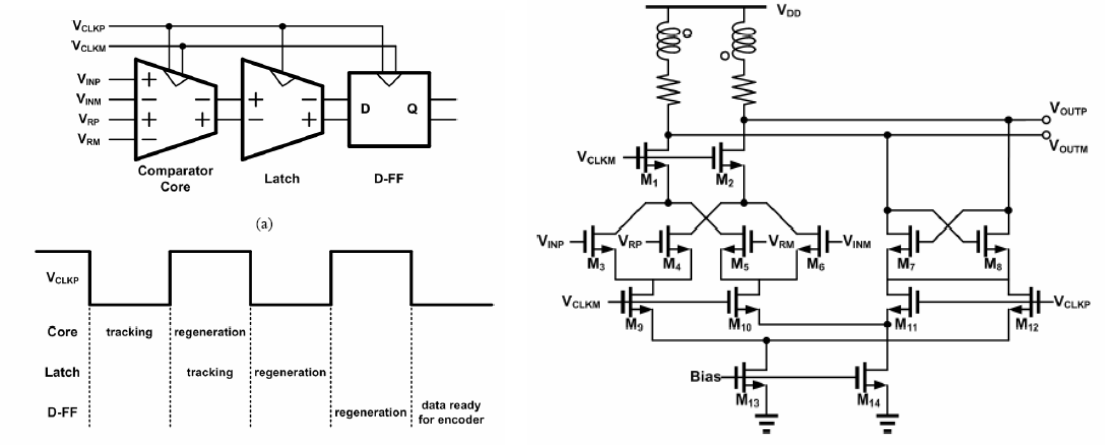
[K.-L.J. Wong and C.-K.K. Yang, "Offset compensation in comparators with minimum input-referred supply noise," JSSC, May 2004.]

High Performance Flash ADC with Calibration (1)



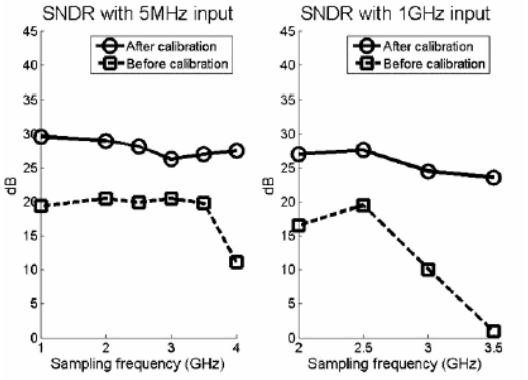
[Park & Flynn, "A 3.5 GS/s 5-b Flash ADC in 90 nm CMOS," CICC 2006]

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)



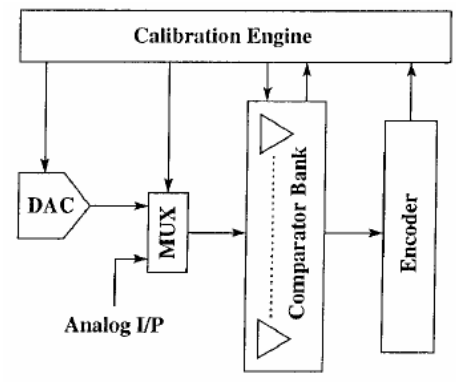
ADC PERFORMANCE

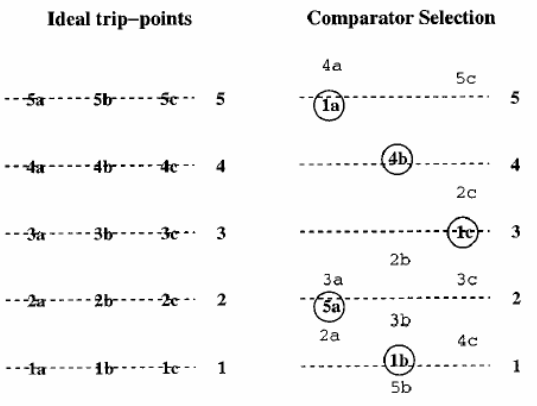
Technology	CMOS 90 nm
Resolution	5 bits
Supply	1.4 V analog, 1.4 V digital, 1.8 V clock buffer
Input range	± 320 mV (LSB = 20 mV)
Sampling rate	Up to 4 GS/s
Power	227 mW (115 mW: comparators, resistor ladder, bias. 17 mW: D-FFs, encoder, decimator. 95 mW: clock buffer)
DNL @ 3.5 GS/s	-0.83 LSB ~ 0.93 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]

Comparator Redundancy (1)

- Idea: Build a "sea of imprecise comparators", then determine which ones to use…





C. Donovan, M. P Flynn, "A 'digital' 6-bit ADC in 0.25-μm CMOS," IEEE J. Solid-State Circuits, pp. 432-437, March 2002.

Comparator Redundancy (2)

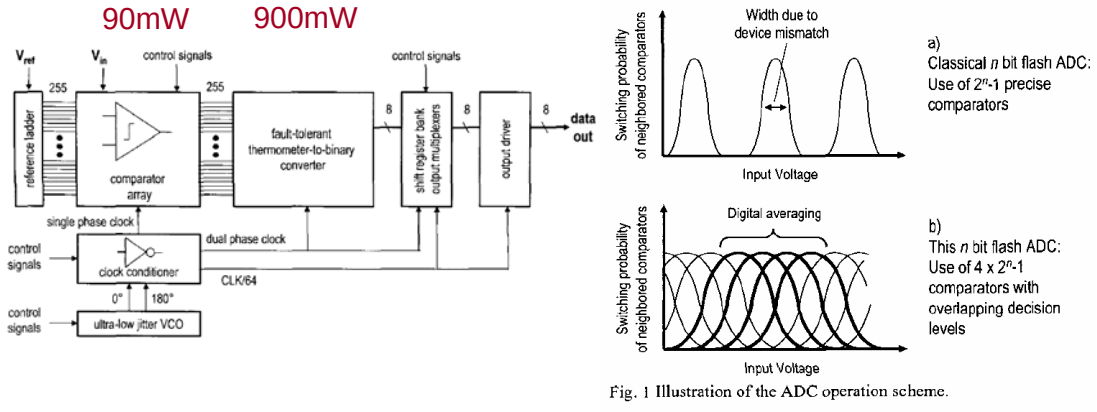
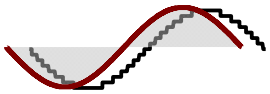


Fig. 1 Illustration of the ADC operation scheme.

Paulus et al., "A 4GS/s 6b flash ADC in 0.13um CMOS," *VLSI Circuits Symposium*, 2004

Lecture 11
Folding & Interpolating ADCs



Boris Murmann
Stanford University
murmann@stanford.edu

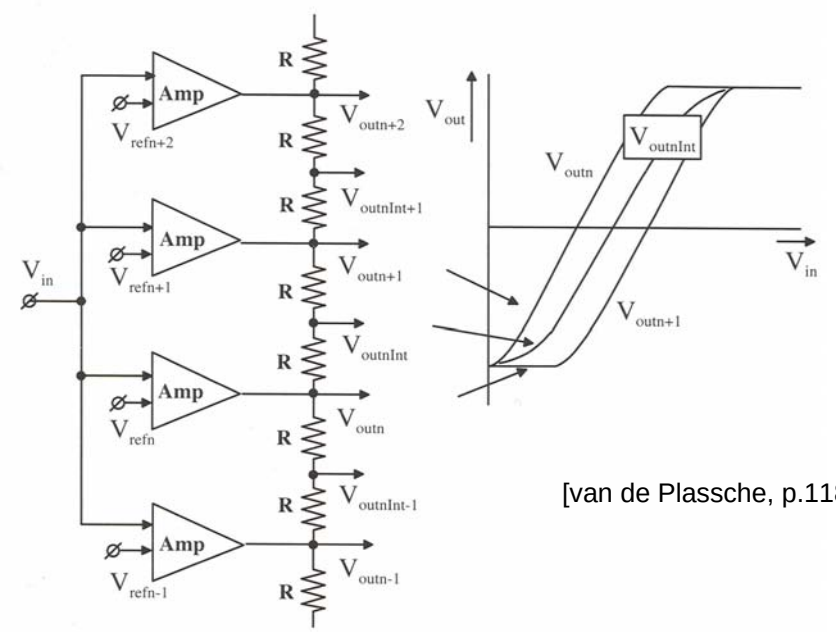
Reducing Complexity

- Example: 10-bit flash ADC
  - Compared to 6-bit example on slide 16, lecture 10, 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

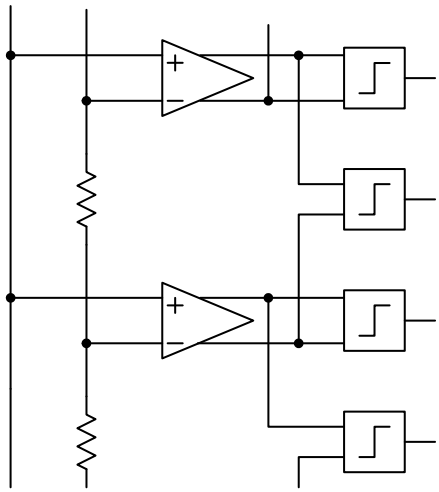
- Idea
  - Interpolation between preamp outputs
- Reduces number of preamps
  - Reduced input capacitance
  - Reduced area, power dissipation
- Same number of latches
- Important “side-benefit”
  - Decreased sensitivity to preamp offset
    - Improved DNL

Concept



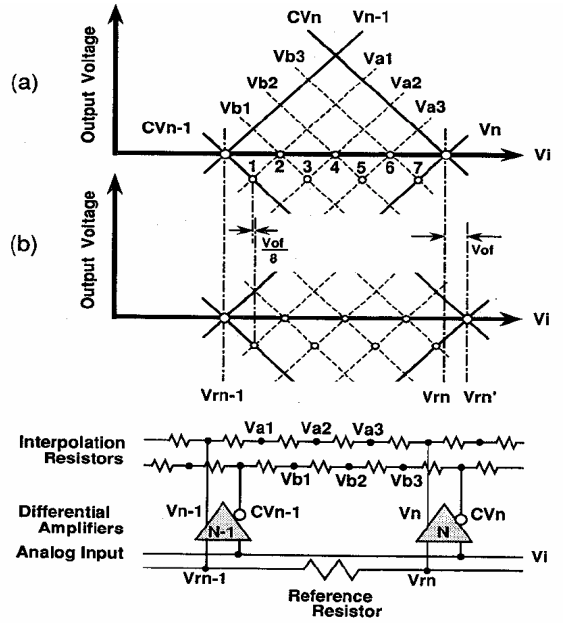
[van de Plassche, p.118]

Differential Implementation



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

Higher Order Interpolation



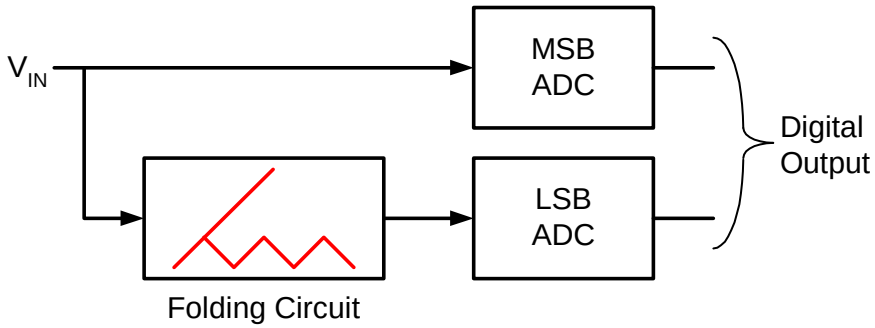
- Resistors produce additional levels
- Define interpolation factor as ratio ratio of latches and preamps
- The example shown on this slide has M=8

[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

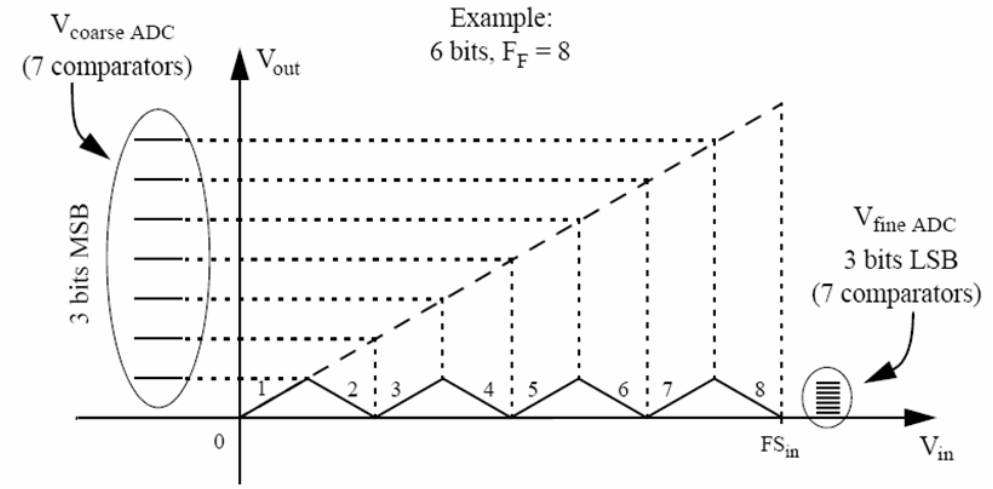
- Must ensure that "linear range" of adjacent preamplifiers overlaps
  - Sets upper bound on preamp gain
- Resistor string reduces signal path bandwidth
  - Sets upper bound on interpolation factor, typically around 4
- For interpolation factors >2, amplifier nonlinearity can limit the precision of zero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

Folding ADC



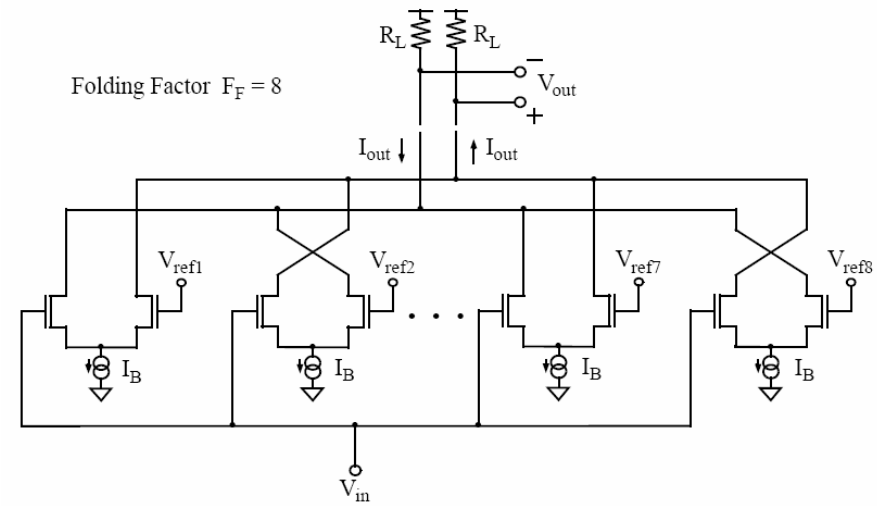
- Fast
- Significantly fewer comparators than flash
- Nonidealities in folder limit attainable resolution to ~10 bits

Example: 6-bit Folding ADC



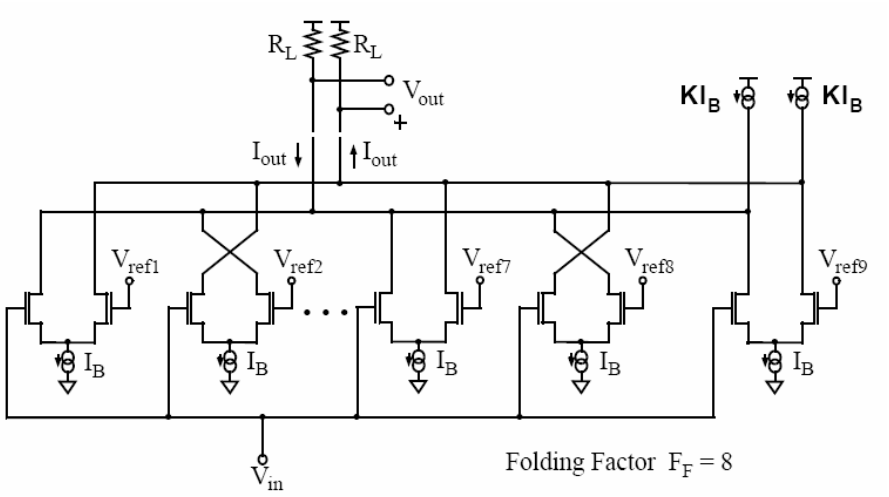
- Coarse ADC determines segment, fine ADC determines level within segment
- Folding factor (F_F) quantifies number of folder output segments

Folder Realization



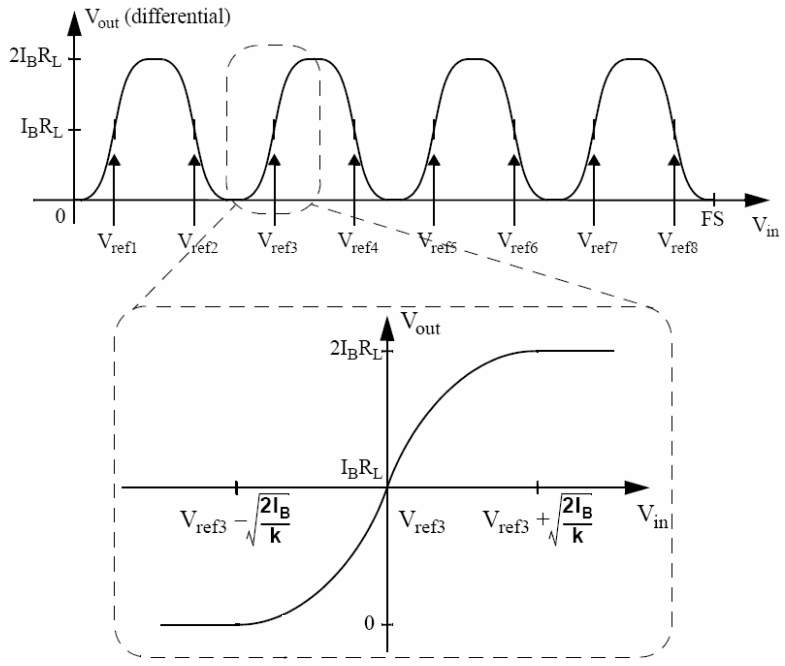
- $V_{ref1} < V_{ref2} < \ldots < V_{ref8}$
- For any V_{in}, only one differential pair is "active", all others are saturated

Improved Realization

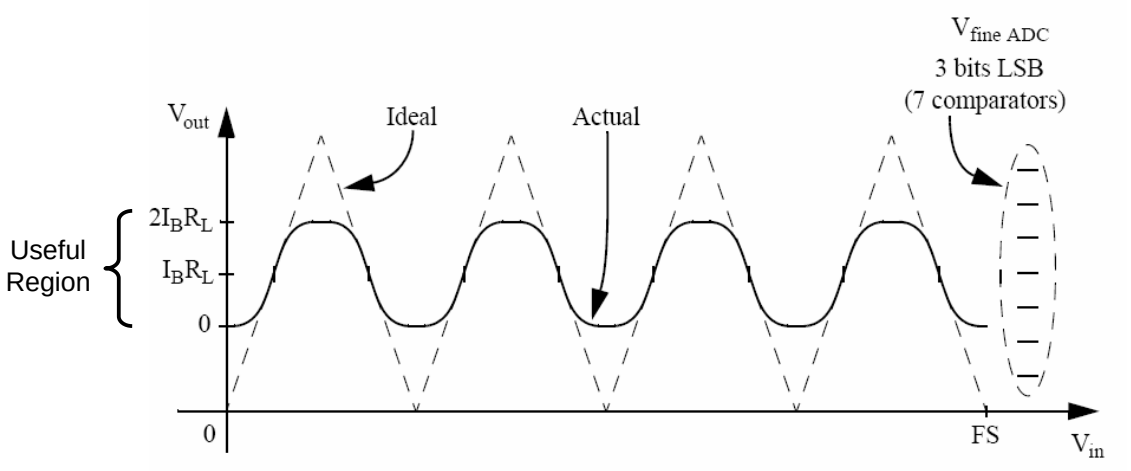


- Extra differential pair removes DC component in V_{out}
- Parameter K controls output common mode

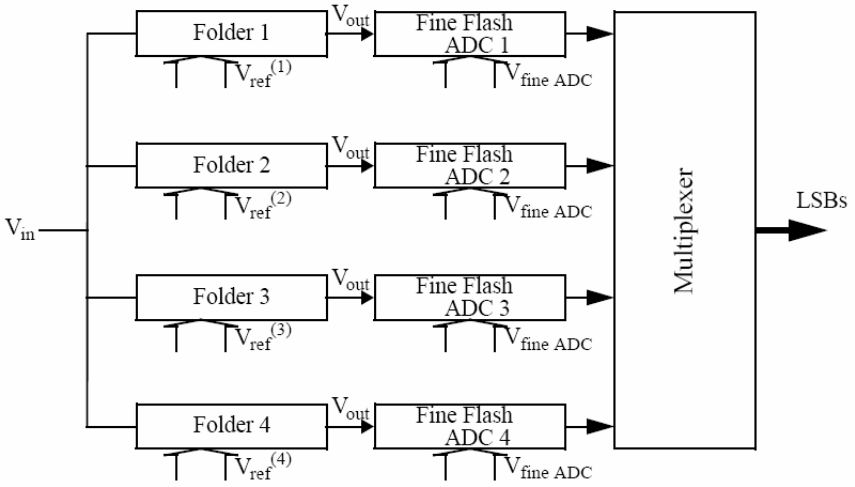
Input-Output Characteristic



Rounding Problem

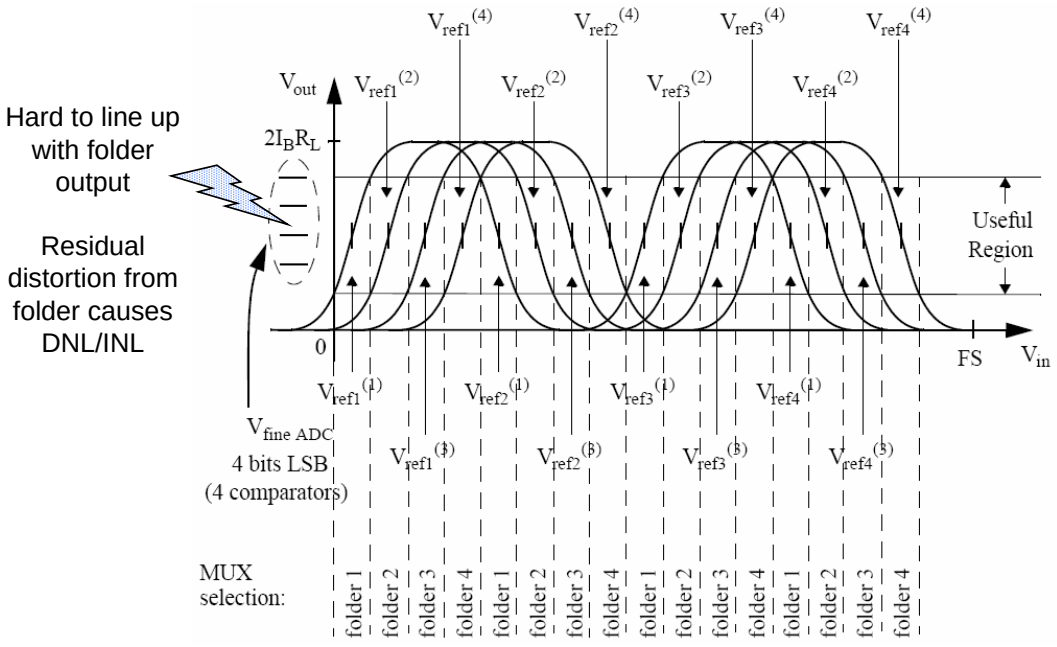


Multiple Folds (2)

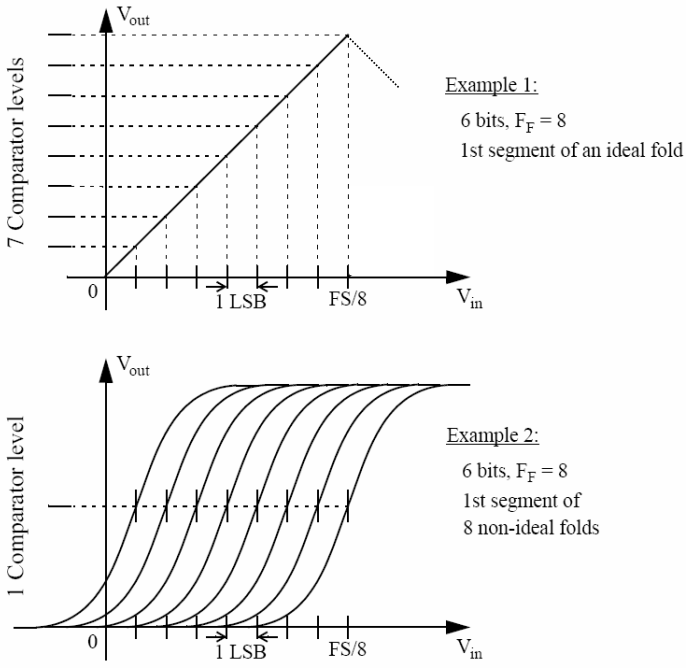


- Idea: Use several folders so that any input value falls into useable "linear" region

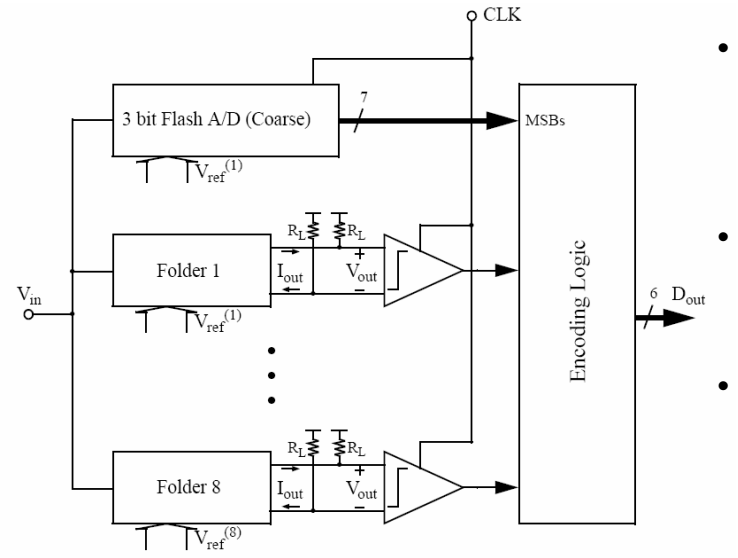
Multiple Folds (2)



Multiple Folds Using Single Threshold (1)

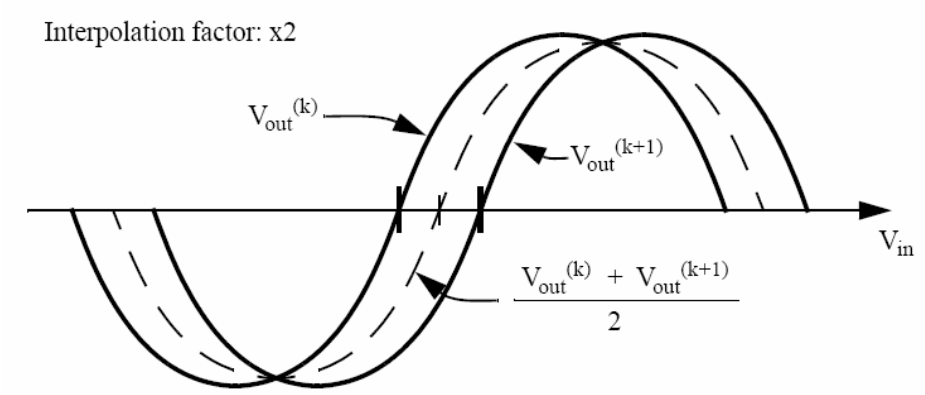


Multiple Folds Using Single Threshold (2)



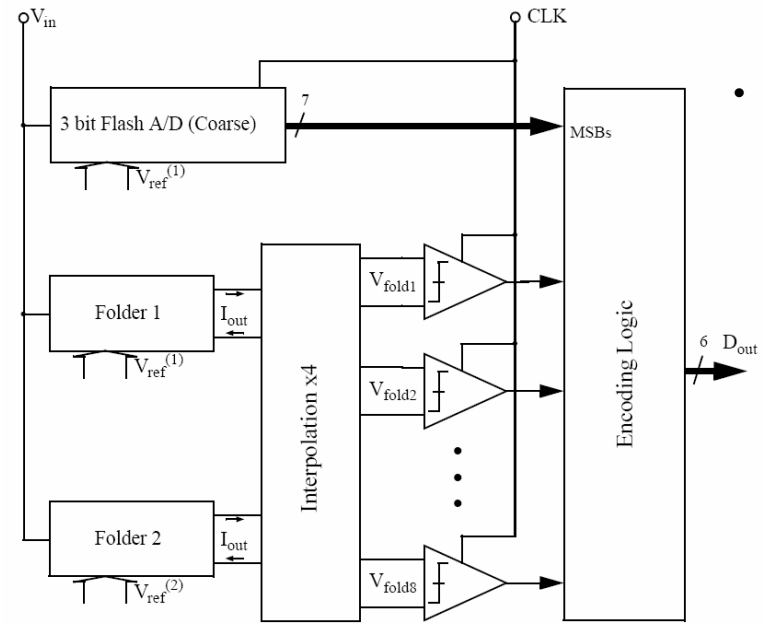
- Initial idea
  - Use one folder and 7 comparators in LSB section
- Now have
  - 8 Folders and 8 comparators (!)
- Yet another idea
  - Use interpolation to eliminate some of the folders

Interpolation



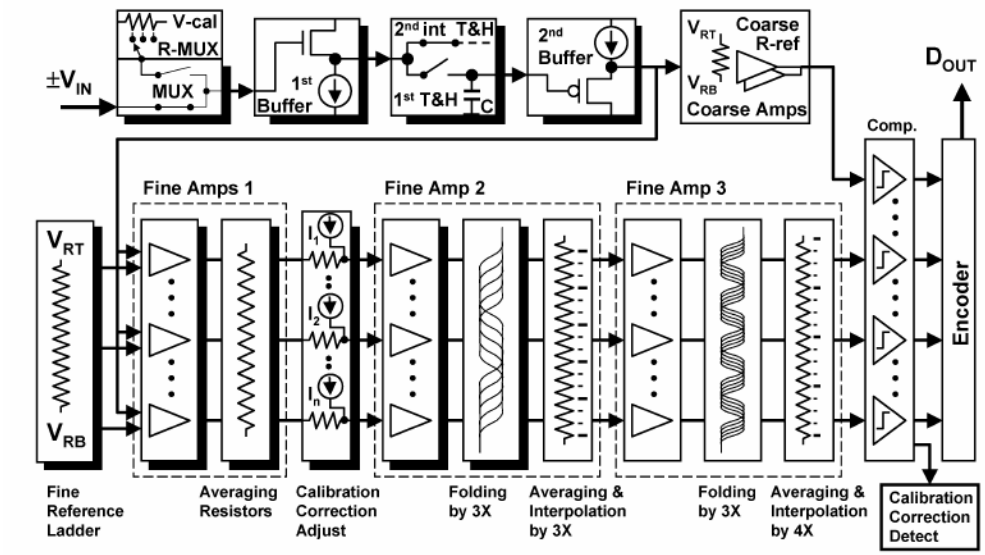
- Same idea as discussed in the context of flash ADCs (slides 3-7)

Complete Folding & Interpolating ADC



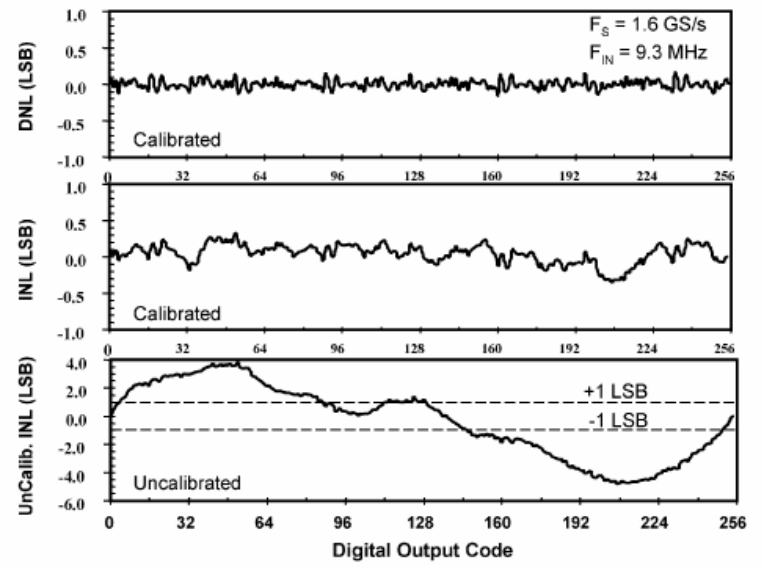
- 6-bit Example
  - 2 folders
  - 4x interpolation
  - F_F=8

State-of-the art Implementation (1)

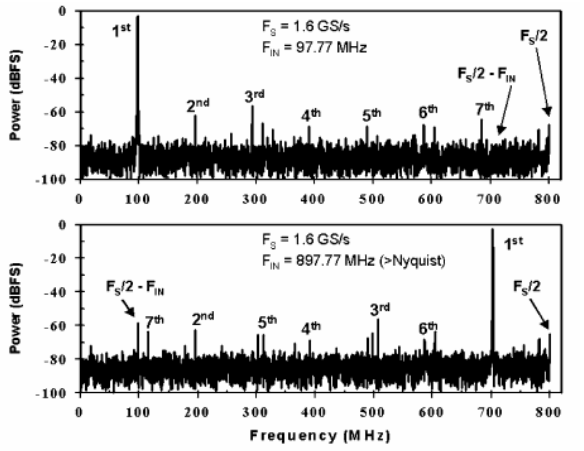


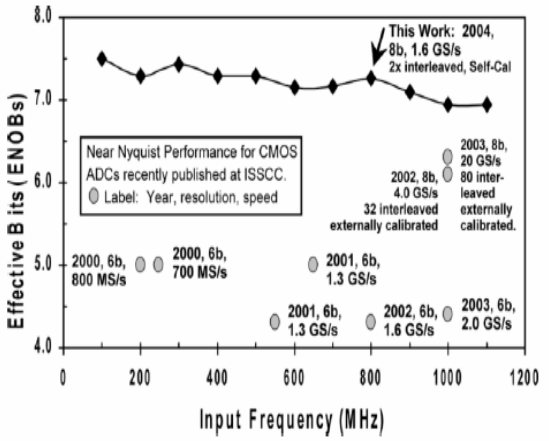
[Taft et al., JSSC 12/2004]

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.

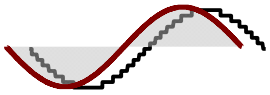
Selected References (2)

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.

Lecture 12
Multi-Step A/D Conversion
Pipeline ADCs

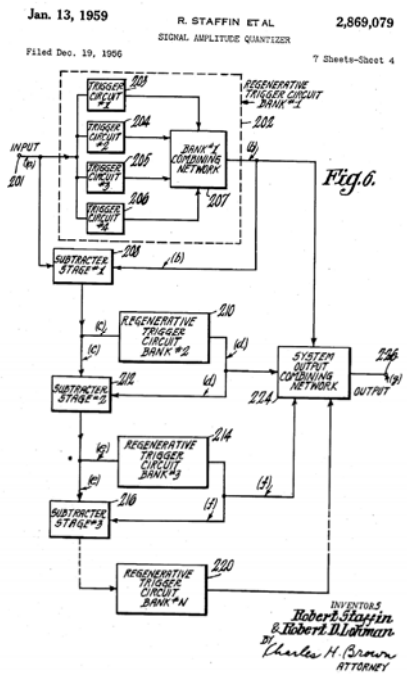


Boris Murmann
Stanford University
murmann@stanford.edu

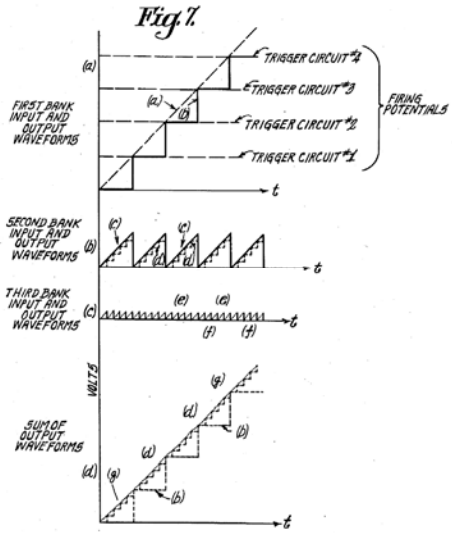
Outline

- 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

History (1)



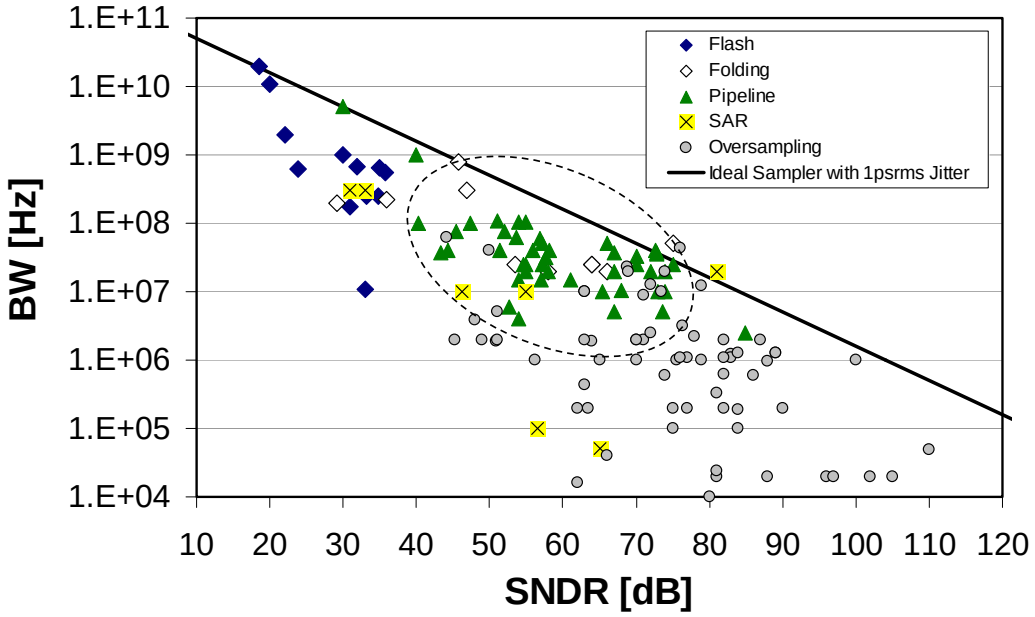
US Patent # 2,869,079: Staffin and Lohman, "Signal Amplitude Quantizer," 1959



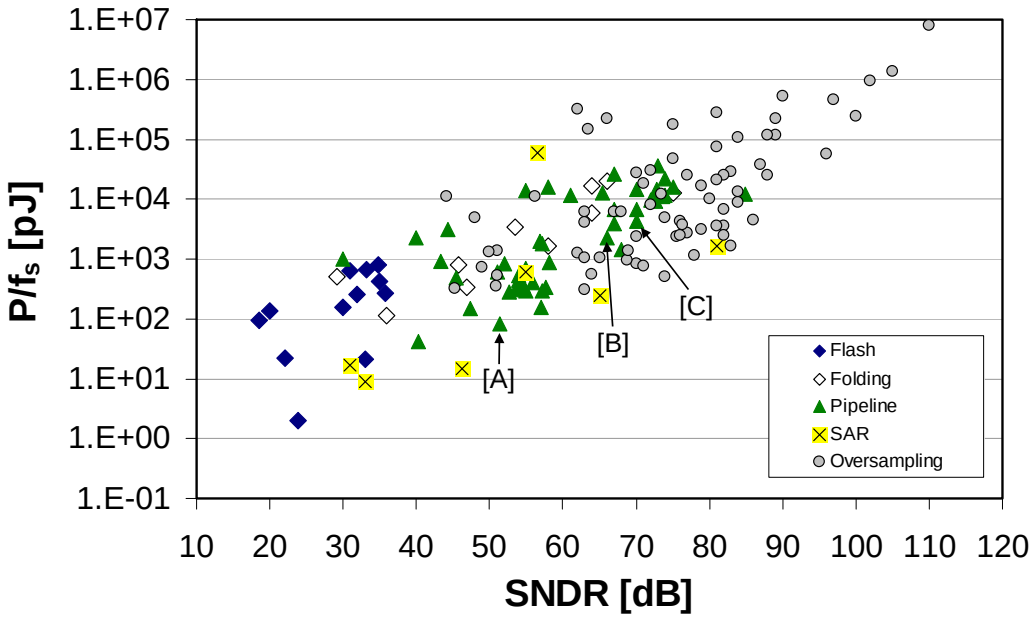
History (2)

- First multi-step ADC with "error correction"
  - T.C. Verster, "A method to increase the accuracy of fast Serial-Parallel Analog-to-Digital Converters," IEEE Trans. Electronic Computers, EC-13, pp. 471-473, 1964.
- First pipeline ADC
  - B.D. Smith, "An Unusual Electronic Analog-Digital Conversion Method," IRE Transactions on Instrumentation, pp.155-160, June 1956.
- First pipeline ADCs in CMOS
  - S.H. Lewis and P.R. Gray, "A pipelined 5-Msample/s 9-bit analog-to-digital converter," JSSC, pp. 954–961, Dec. 1987.
  - S. Sutarja and P.R. Gray, "A pipelined 13-bit 250-ks/s 5-V analog-to-digital converter," JSSC, pp. 1316–1323, Dec. 1988.

ADC Performance Survey (ISSCC 1997-2007)



ADC Performance Survey (ISSCC 1997-2007)

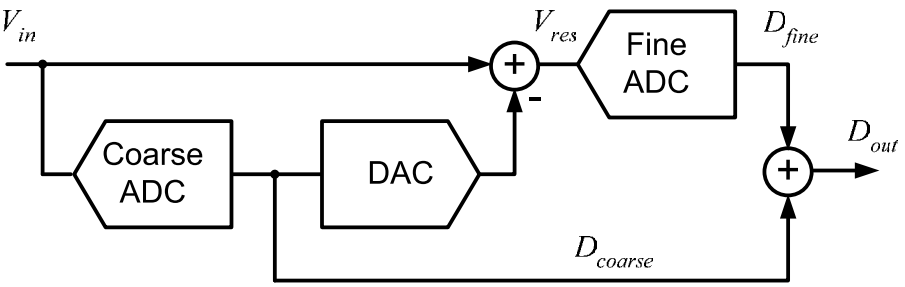


State-of-the-Art Examples

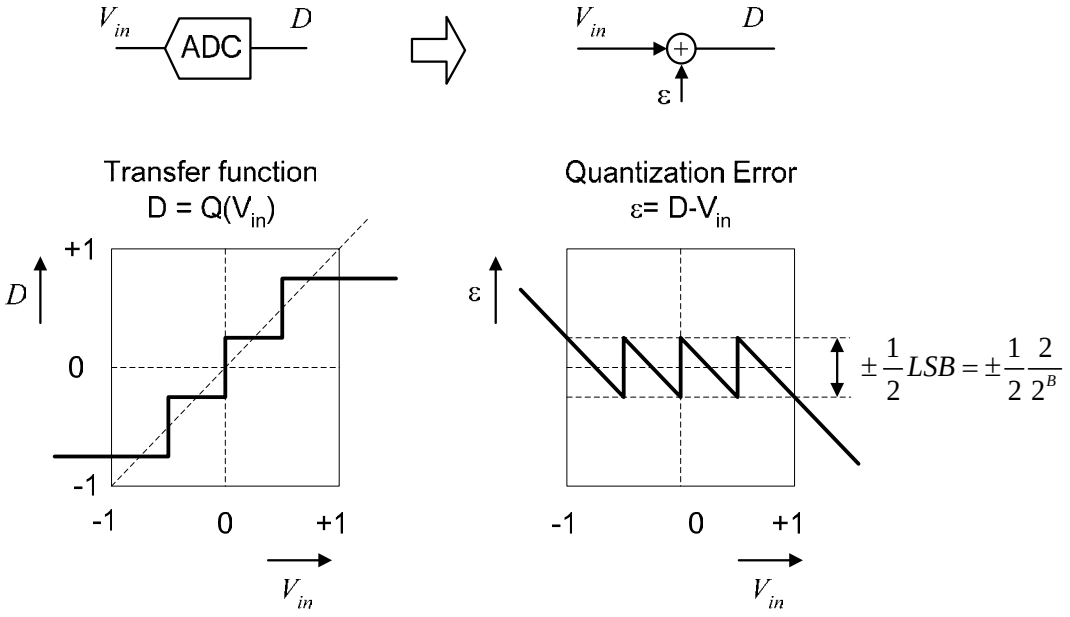
- **[A]** M. Yoshioka et al., "A 0.8V 10b 80MS/s 6.5mW Pipelined ADC with Regulated Overdrive Voltage Biasing," ISSCC Dig. Techn. Papers, pp. 452-453, Feb. 2007.
  - **8.3 ENOB** @ Nyquist, **0.08mW per MS/s**, 90nm CMOS (0.8V)
  - **9.0 ENOB** @ Nyquist, **0.16mW per MS/s**, 90nm CMOS (1.2V)
- **[B]** 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.
  - **10.7 ENOB** @ Nyquist, **2.24mW per MS/s**, 0.13um CMOS
- **[C]** 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.
  - **11.8 ENOB** @ Nyquist, **4.53mW per MS/s**, 0.35um CMOS

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

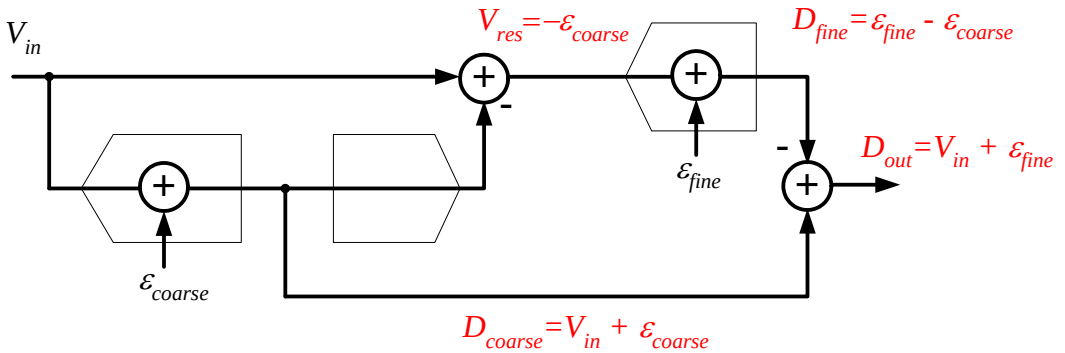


Quantizer Model



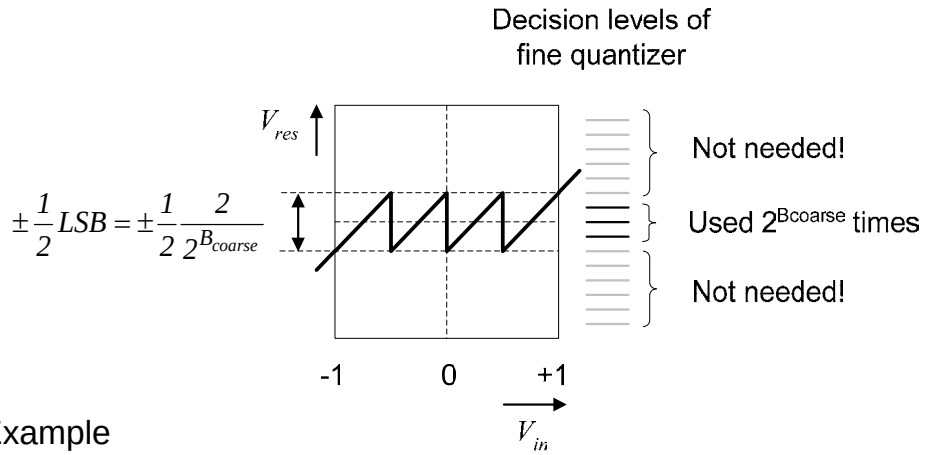
Analysis

- Assuming ideal DAC (for now)



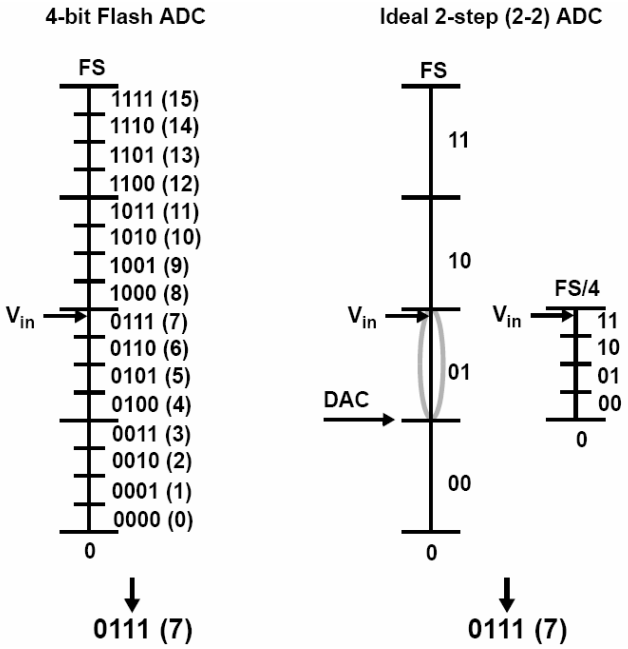
- Output contains only quantization error from fine ADC!

Input to Fine Quantizer (V_{res})

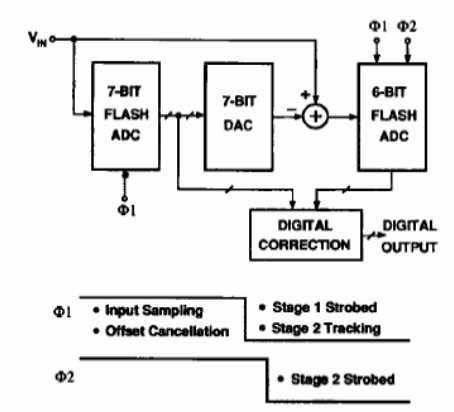


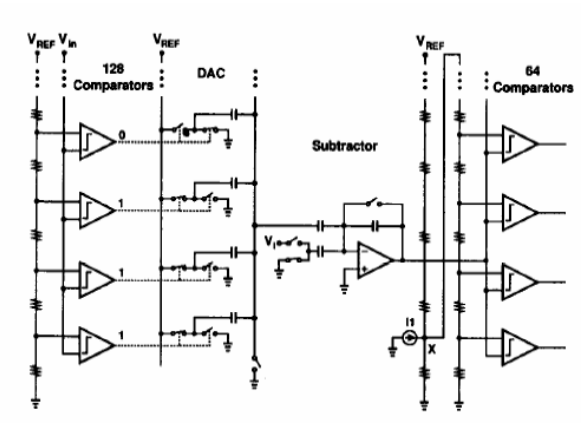
- Example
  - Three decision levels in coarse and fine quantizer
  - Aggregate ADC resolution is 4 bits (3+4·3 decision levels)
  - Need only 6 comparators, compared to 15 in a 4-bit flash ADC
    - Advantage becomes more pronounced at higher resolutions

Alternative Illustration



Two-Step ADC Example

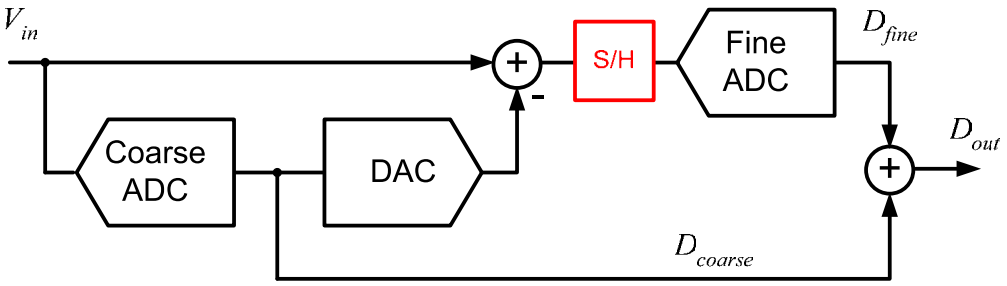




[B. Razavi and B.A. Wooley, "A 12-b 5-Msample/s two-step CMOS A/D converter," IEEE JSSC, pp. 1667-1678, Dec. 1992.

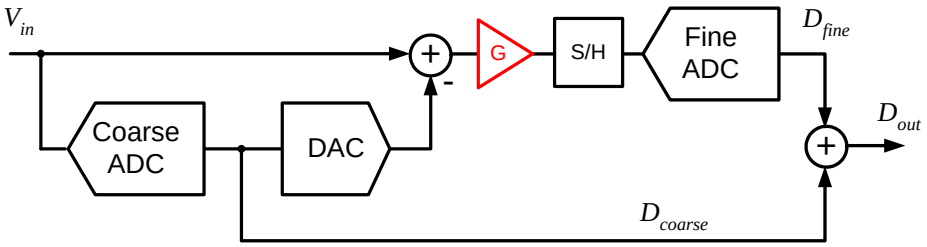
Limitations (1)

- 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"

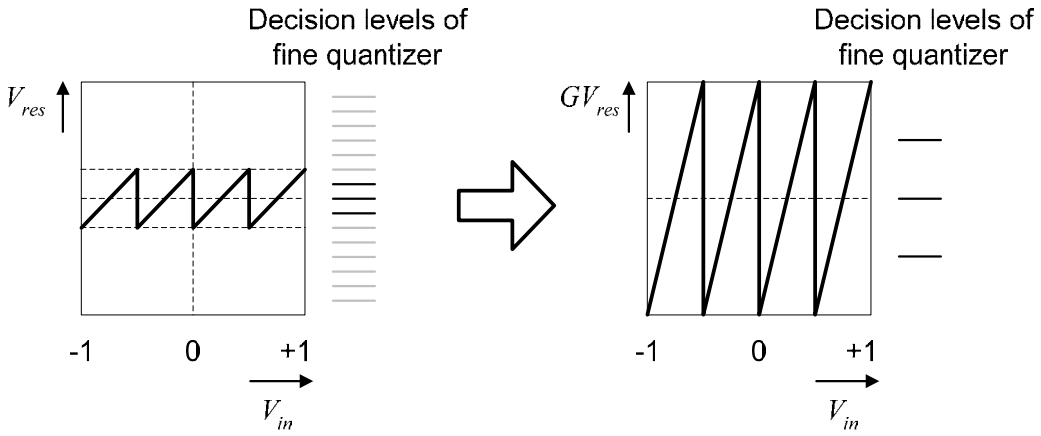


Limitations (1)

- 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

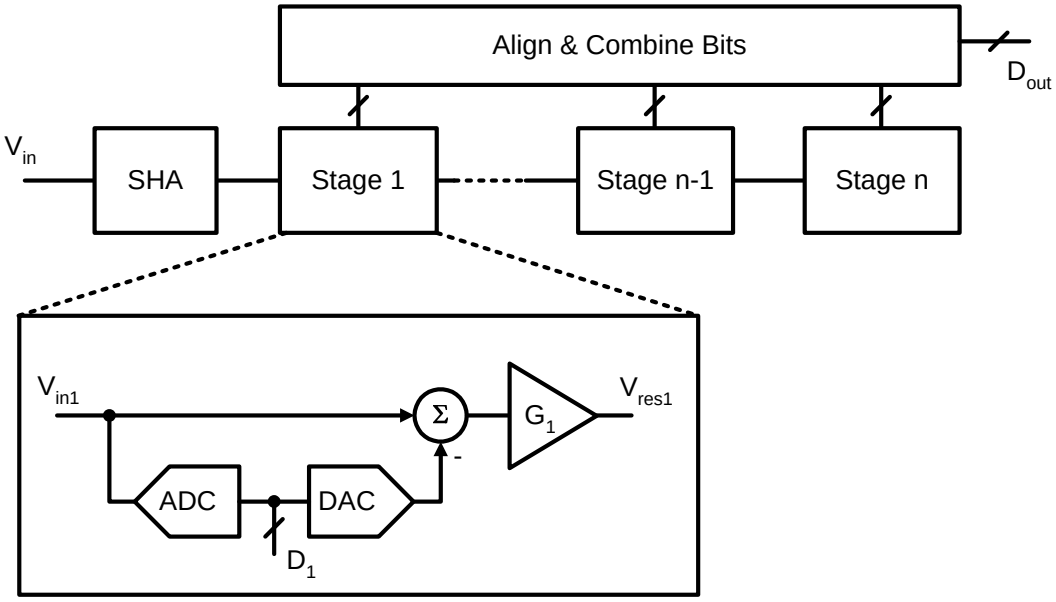


Input to Fine Quantizer with Gain

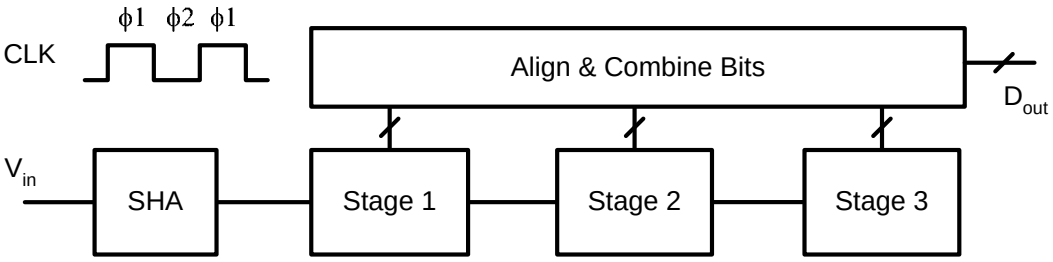


- No longer need precision comparators

Pipeline ADC Block Diagram



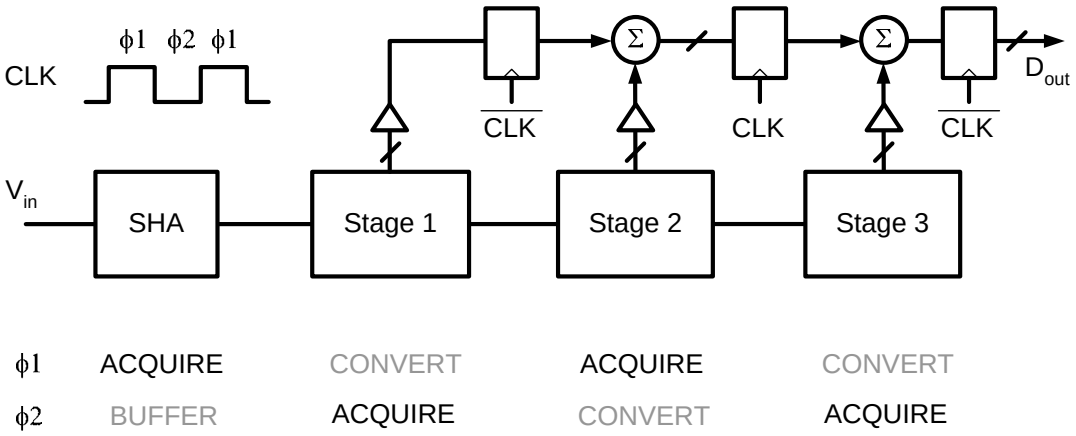
Concurrent Stage Operation



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

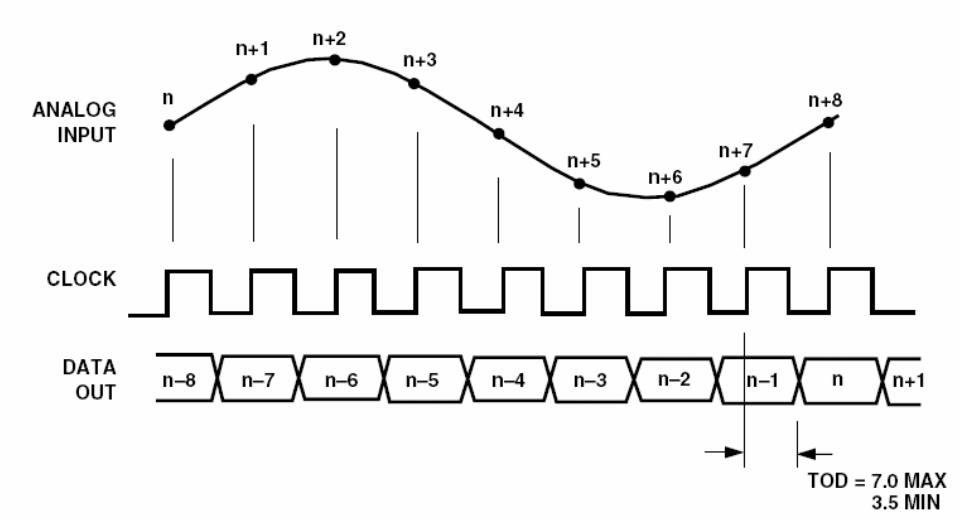
Data Alignment



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



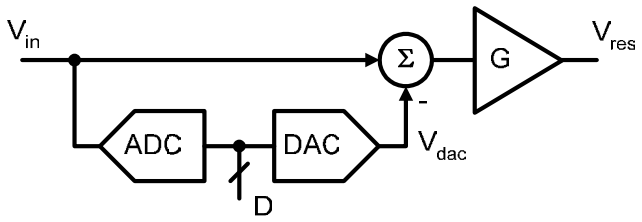
[Analog Devices, AD9226 Data Sheet]

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

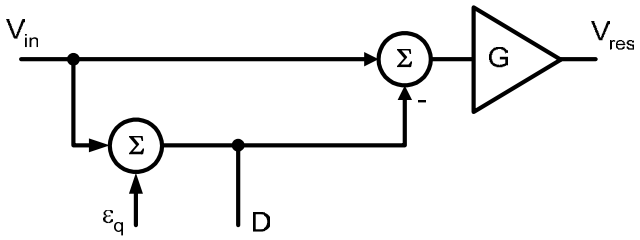
- Ignore timing/clock delays for simplicity



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

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

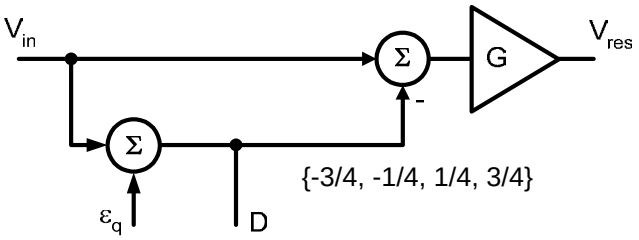
Stage Model with Ideal DAC

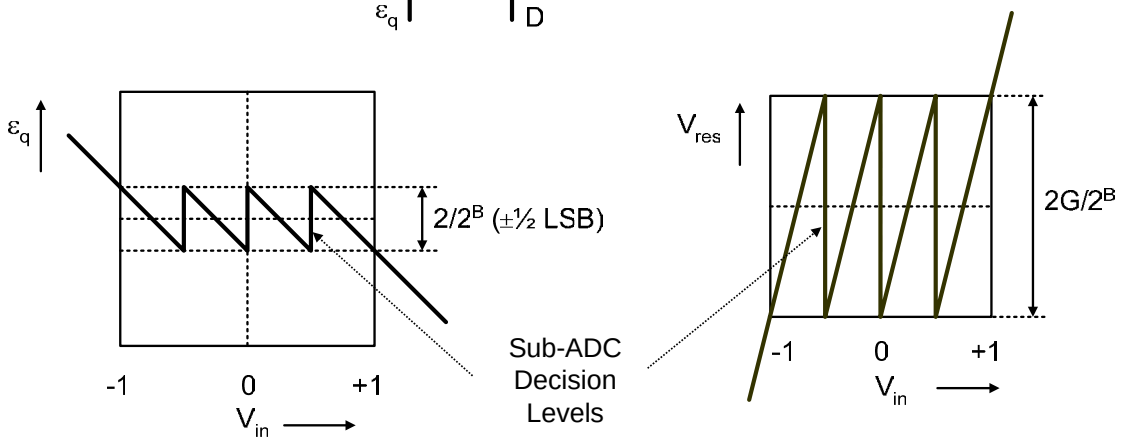


$$D = V_{in} + \varepsilon_q \qquad V_{res} = -G \cdot \varepsilon_q$$

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

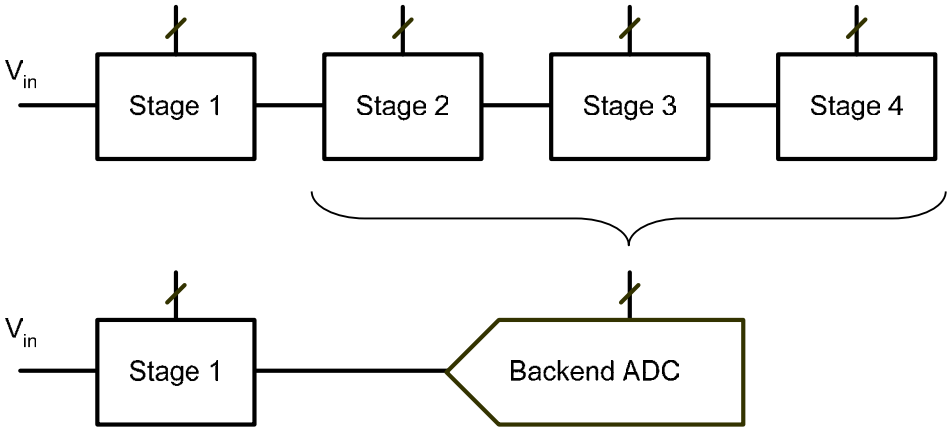
"Residue Plot" (2-bit Sub-ADC)



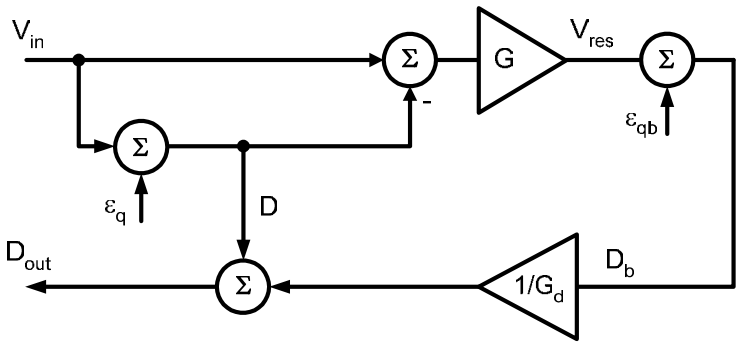


Pipeline Decomposition

- Often convenient to look at pipeline as single stage plus backend ADC



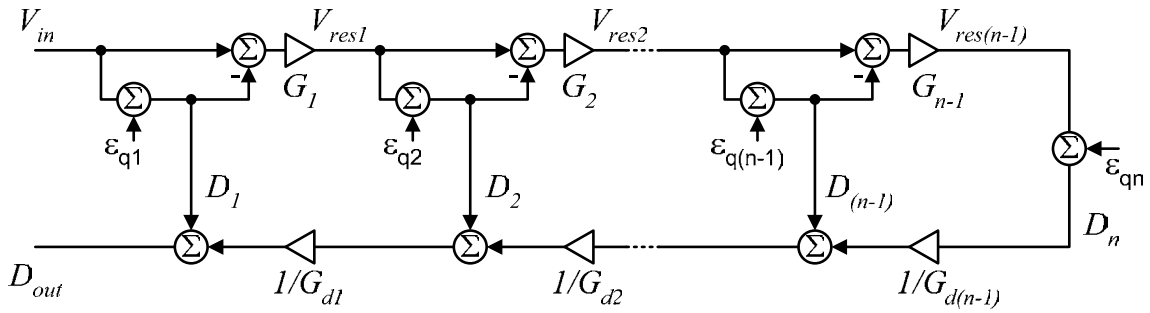
Resulting Model



$$D_{out} = V_{in} + \varepsilon_q \left(1 - \frac{G}{G_d}\right) + \frac{\varepsilon_{qb}}{G_d}$$

With $G_{d1}=G_1$

Canonical Extension



$$D_{out} = V_{in} + \varepsilon_{q1}\left(1 - \frac{G_1}{G_{d1}}\right) + \frac{\varepsilon_{q2}}{G_{d1}}\left(1 - \frac{G_2}{G_{d2}}\right) + \ldots + \frac{\varepsilon_{q(n-1)}}{\prod_{j=1}^{n-2} G_{dj}}\left(1 - \frac{G_{(n-1)}}{G_{d(n-1)}}\right) + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_{dj}}$$

- First stage has most stringent precision requirements
- Note that above model assumes that all stages use same reference voltage (same full scale range)
  - This is true for most designs, one exception is [Limotyrakis 2005]

General Result – Ideal Pipeline ADC

- With ideal DACs and ideal digital weights ($G_{dj}=G_j$)

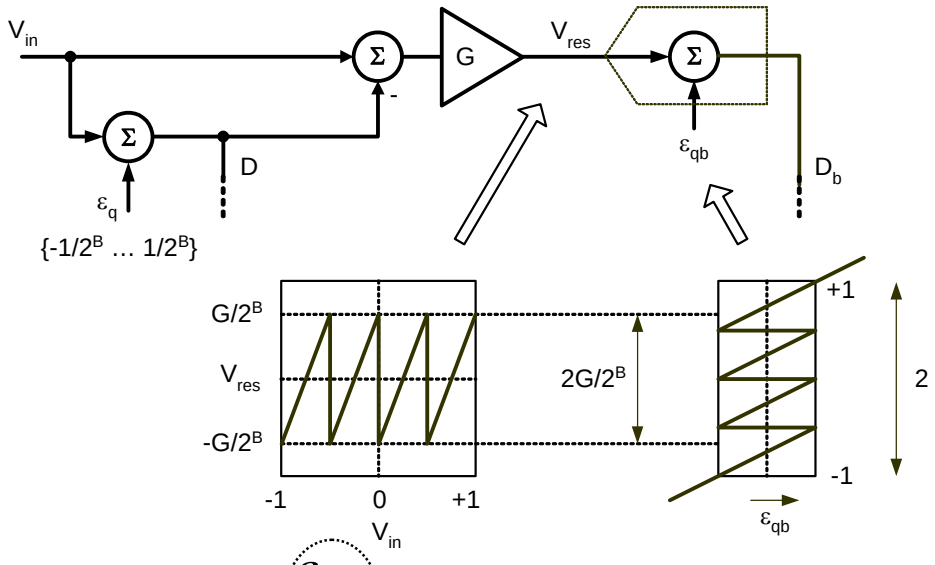
$$D_{out} = V_{in} + \frac{\varepsilon_{qn}}{\prod_{j=1}^{n-1} G_j} \quad \Rightarrow \quad B_{ADC} = B_n + \sum_{j=1}^{n-1} log_2 G_j$$

- The only error in D_{out} is that of last quantizer, divided by aggregate gain
- Aggregate ADC resolution is independent of sub-ADC resolutions in stage 1...n-1 (!)
- Makes sense to define "effective" resolution of j^{th} stage as $R_j=\log_2(G_j)$

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

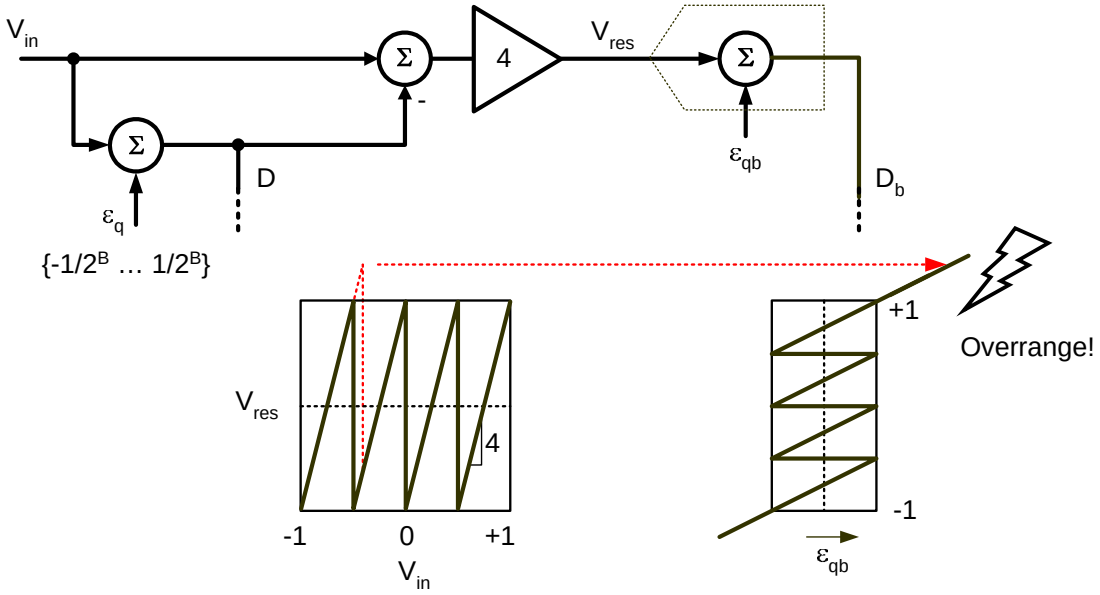
Upper Bound for Stage Gain



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

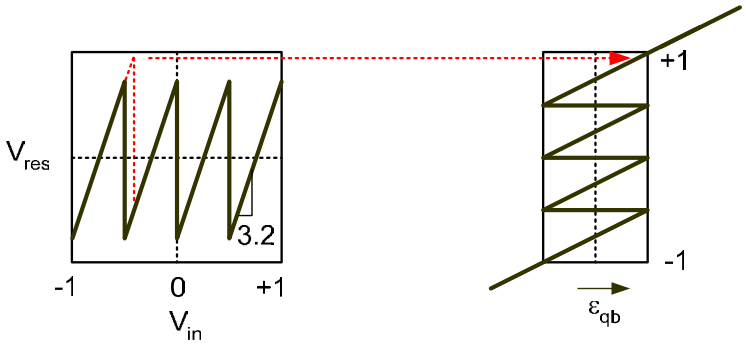
Grows out of ±½ LSB bounds for $G>2^B$

Issue with $G=2^B$



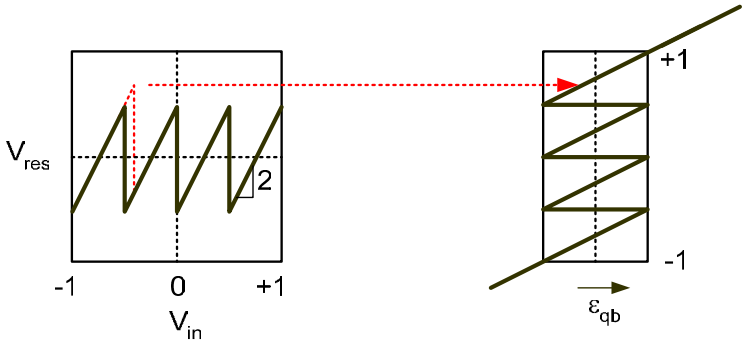
- Any error in sub-ADC decision levels will overload backend ADC and thereby deteriorate ADC transfer function

Idea #1: G slightly less than 2^B



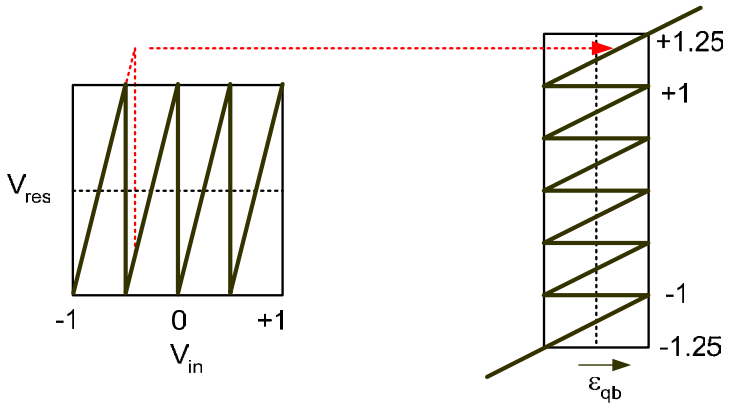
- Effective stage resolution can be non-integer ($R=\log_2 G$)
  - E.g. $R = \log_2 3.2 = 1.68$ bits
- See e.g. [Karanicolas 1993]

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



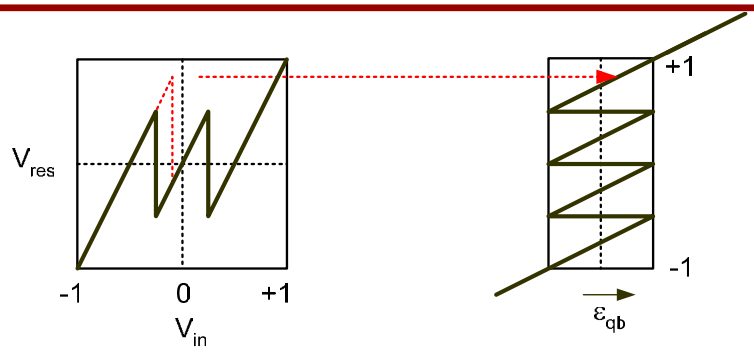
- Effective stage resolution is an integer
  - E.g. R = $\log_2 2 = 1 = B-1$
  - Digital hardware requires only a few adders, no need to implement fractional weights (see appendix)
- See e.g. [Mehr 2000]

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"

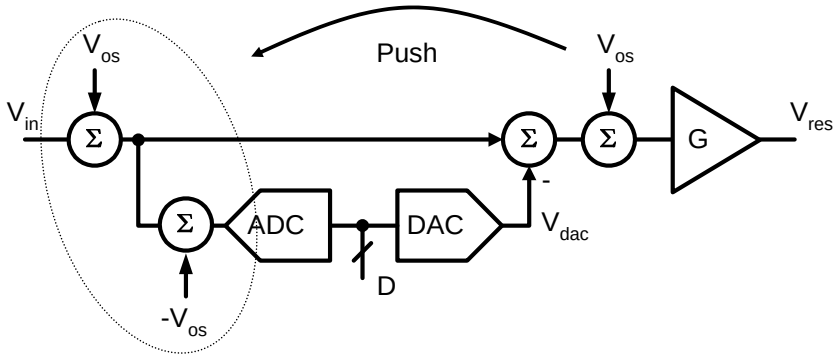


- Sub-ADC decision levels placed to minimize comparator count
- Can accommodate errors up to ±¼
- B = $\log_2(2+1)$ = 1.589 (sub-ADC resolution)
- R = $\log_2 2$ = 1 (effective stage resolution)
- See e.g. [Lewis 1992]

Summary on Sub-ADC Redundancy

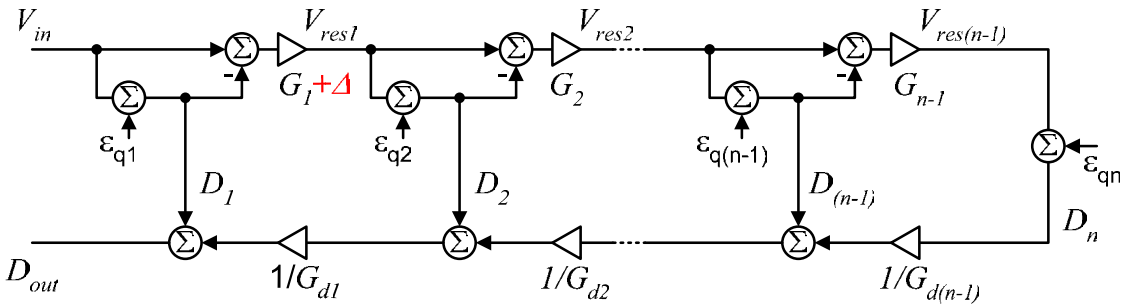
- We can tolerate sub-ADC errors as long as
  - The residue stays "inside the box", or
  - Another stage downstream returns the residue "into the box" before it reaches last quantizer
- This result applies to any stage in an n-stage pipeline
  - Can always decompose pipeline into single stage + backend 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



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

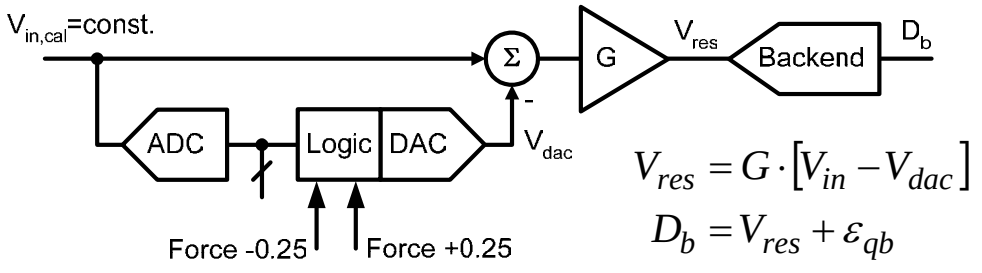
- Want to make $G_{d1} = G_1 + \Delta$

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)

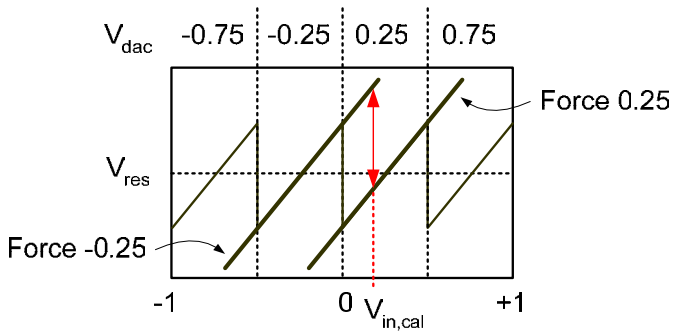
- Note
  - Even if all G_{dj} are perfectly adjusted to reflect the analog gains, the ADC will have non-zero DNL and INL, bounded by ±0.5LSB. This can be explained by the fact that the residue transitions may not correspond to integer multiples of the backend-LSB. This can cause non-uniformity in the 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)

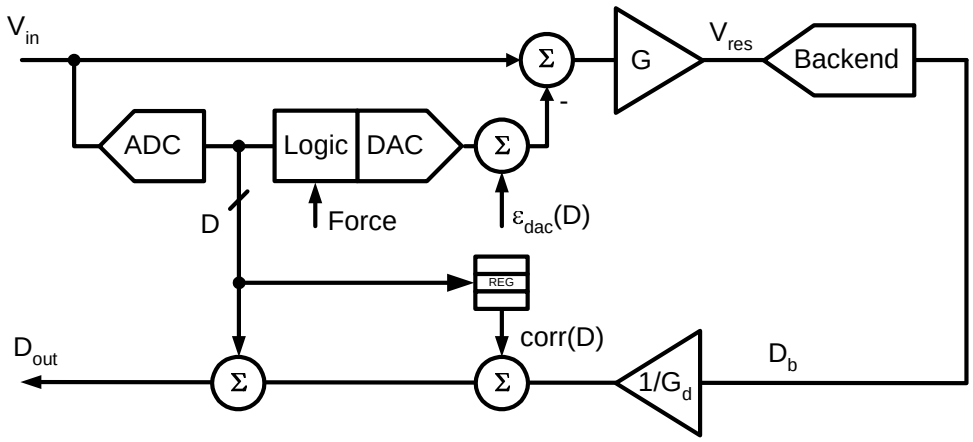
Step1: $$D_b^{(1)} = G \cdot [V_{in} + 0.25] + \varepsilon_{qb}^{(1)}$$

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

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

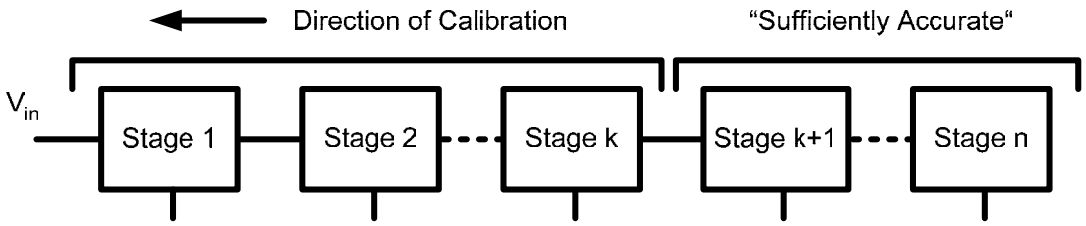
- Can minimize impact of quantization error using
  - Averaging (thermal noise dither)
  - Extra backend resolution

DAC Calibration



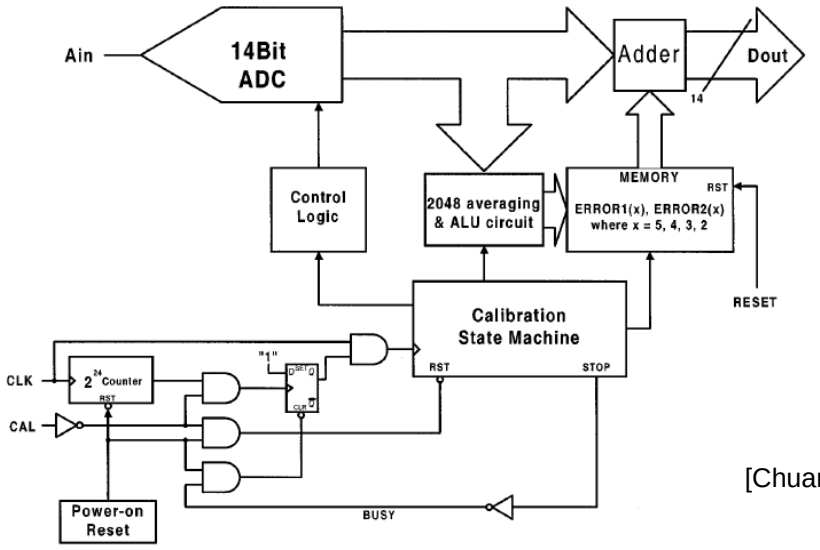
- Essentially same concept as gain calibration
  - Step through DAC codes and use backend to measure errors
- Store coefficients for each DAC transition in a look-up table

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

Calibration Hardware Example



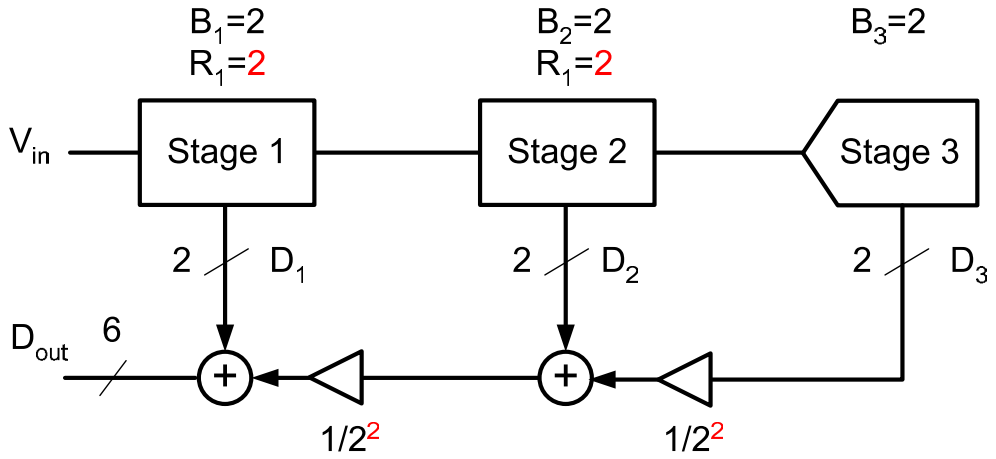
[Chuang 2002]

Alternative Schemes

- Other foreground calibration schemes
  - Calibrate ADC starting from first stage [Singer 2000]
  - Connect stages in a circular loop [Soenen 1995]
- Background calibration
  - See e.g. [Ming 2001]
  - Makes sense primarily when calibration parameters are expected to drift
    - Capacitor ratios do not drift!
    - Background calibration is justifiable e.g. when drift in OTA open-loop gain is an issue

Combining the Bits (1)

- Example1: Three 2-bit stages, no redundancy



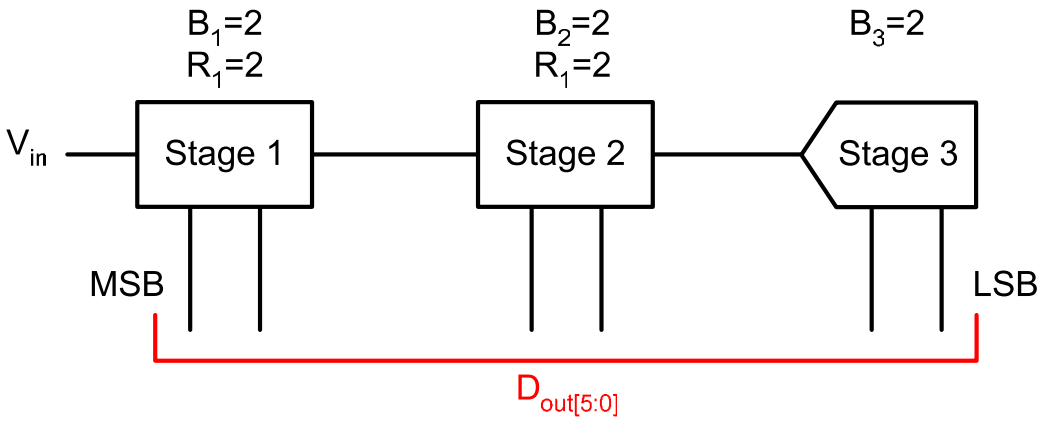
$$D_{out} = D_1 + \frac{1}{4}D_2 + \frac{1}{16}D_3$$

Combining the Bits (2)

D_1 XX
D_2 XX
D_3 XX

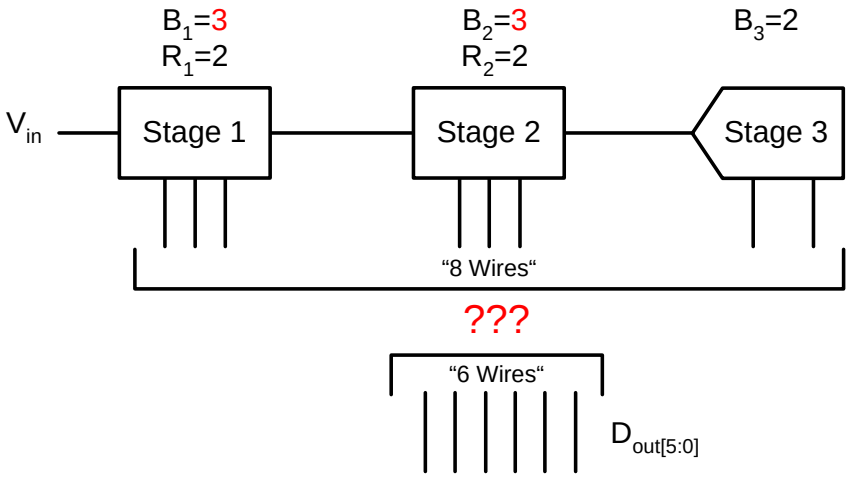
D_{out} DDDDDD

- 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)

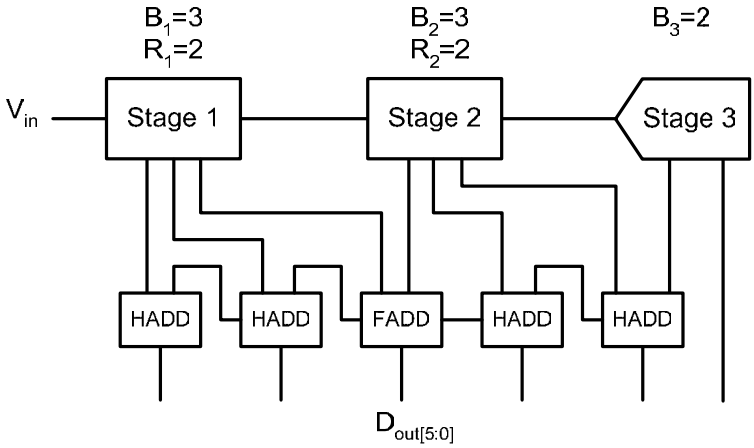


Combining the Bits (4)

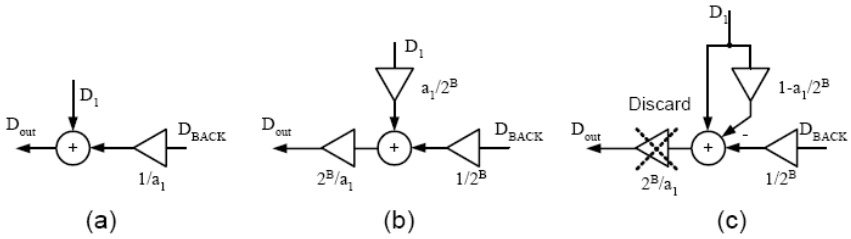
$$D_{out} = D_1 + \frac{1}{4}D_2 + \frac{1}{16}D_3$$

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

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

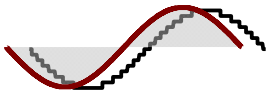


Combining the Bits (5)



- 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]

Lecture 13
Pipeline ADCs (Continued)

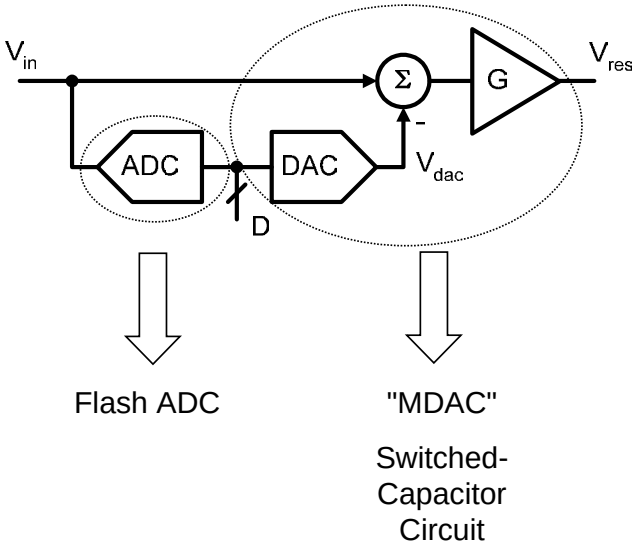


Boris Murmann
Stanford University
murmann@stanford.edu

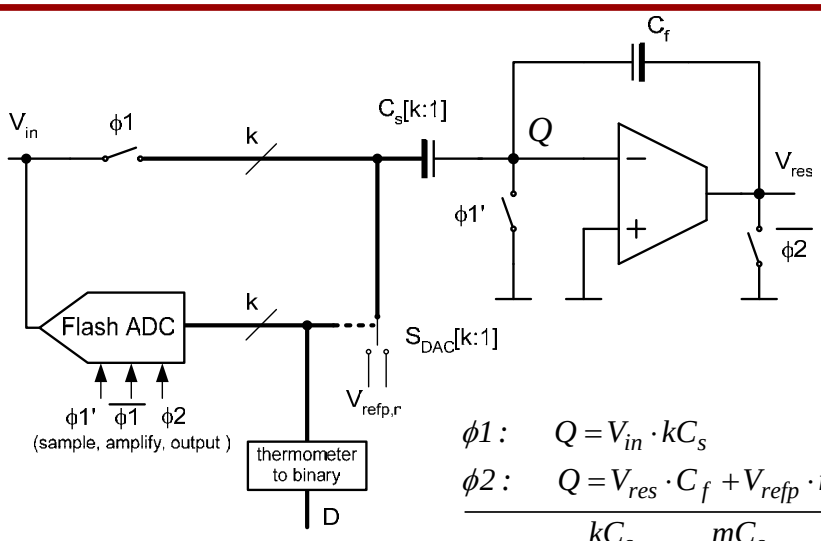
Outline

- 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



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

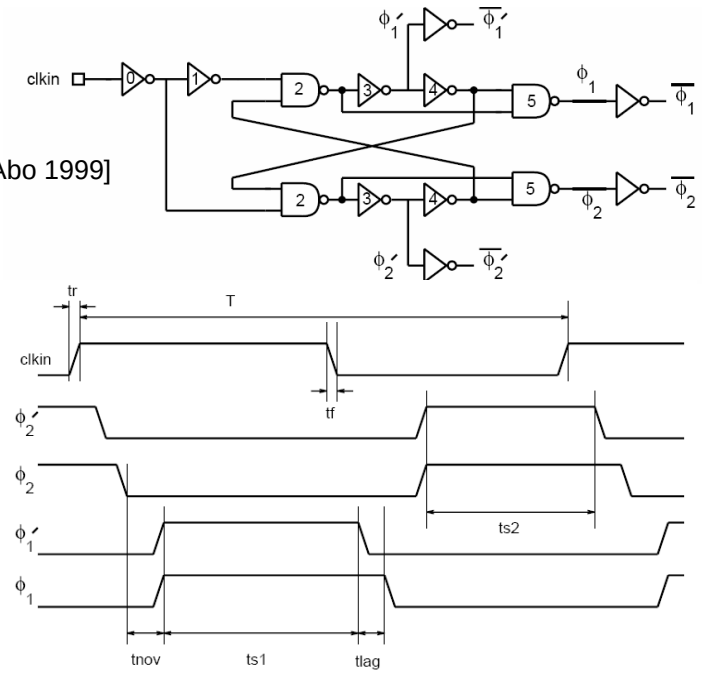
$$\phi 2: \quad Q = V_{res} \cdot C_f + V_{refp} \cdot mC_s + V_{refn} \cdot (k-m)C_s$$

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

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

Generation of Non-Overlapping Clocks

See e.g. [Abo 1999]



OK to make t_{nov}, t_{lag}~100…200ps in today's technology

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)

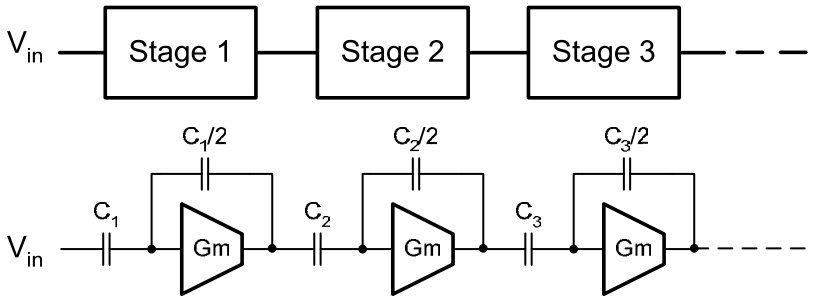
➢ A very complex optimization problem!

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)

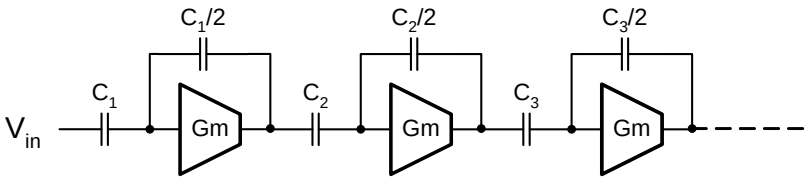
- Example: Pipeline using 1-bit (effective) stages (G=2)



- Total input referred noise power

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

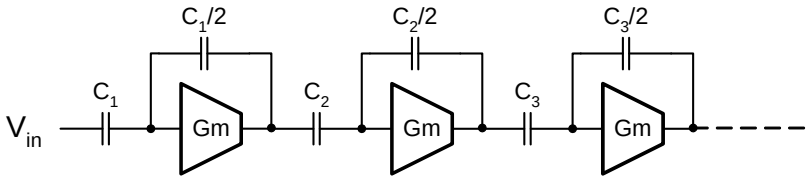
Stage Scaling (2)



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

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

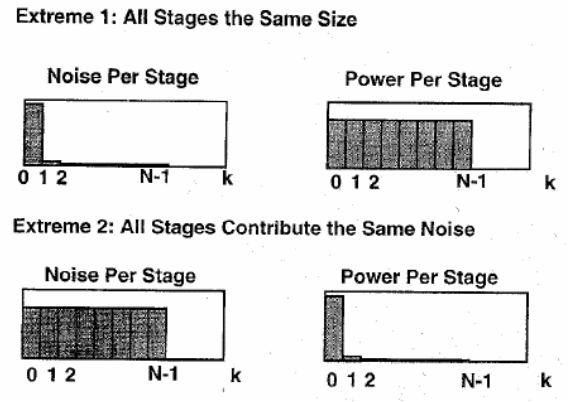
Stage Scaling (3)



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

- 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)

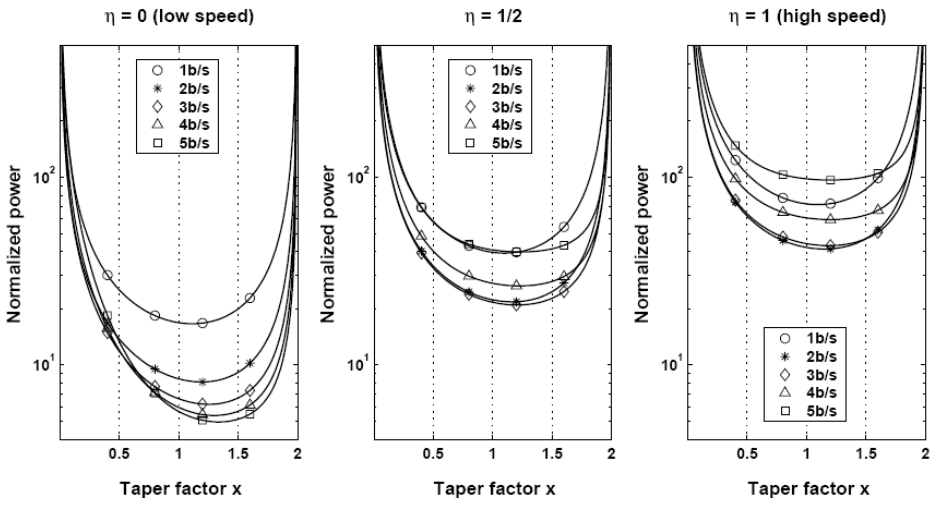


[Cline 1996]

- Optimum capacitior scaling lies approximately midway between these two extremes

Shallow Optimum

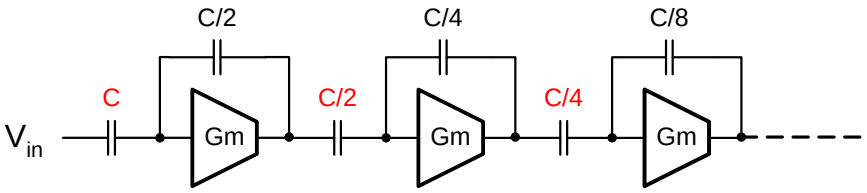
[Chiu 2004]



Capacitor scaling factor = 2^{Rx} $x=1 \Rightarrow$ scaling exactly by stage gain

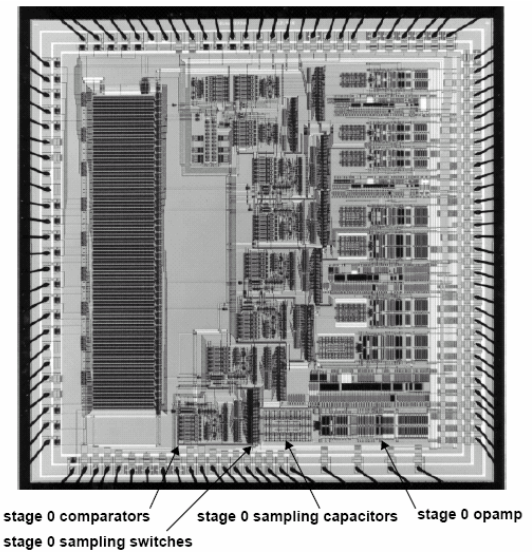
Practical Approach to Stage Scaling

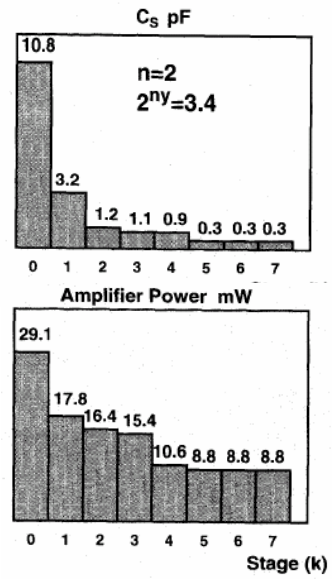
- Start by assuming caps are scaled precisely by stage gain
  - E.g. for 1-bit effective stages:



- 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…

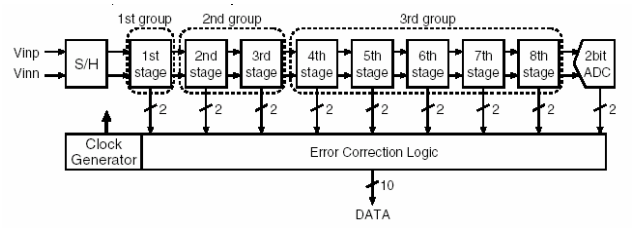
Stage Scaling Examples (1)

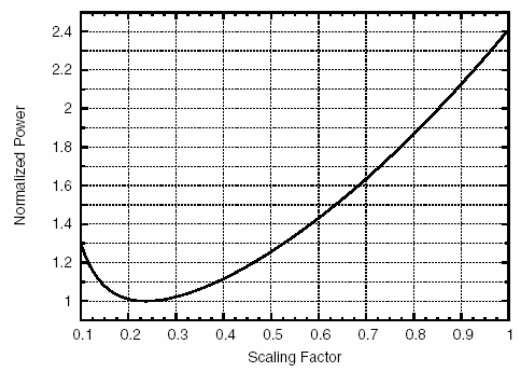




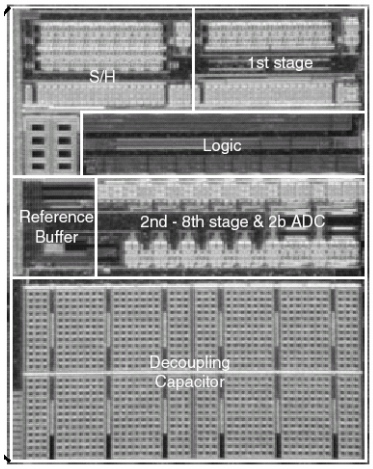
[Cline 1996]

Stage Scaling Examples (2)





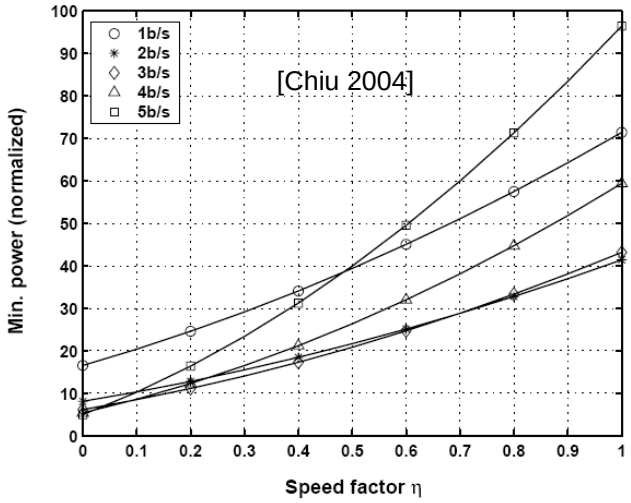
[Ishii 2005]



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

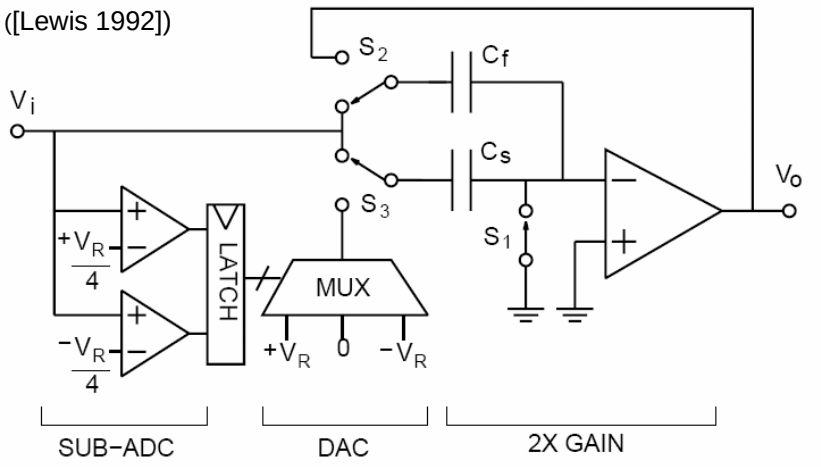
- Low power is possible for a wide range of architectures!

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

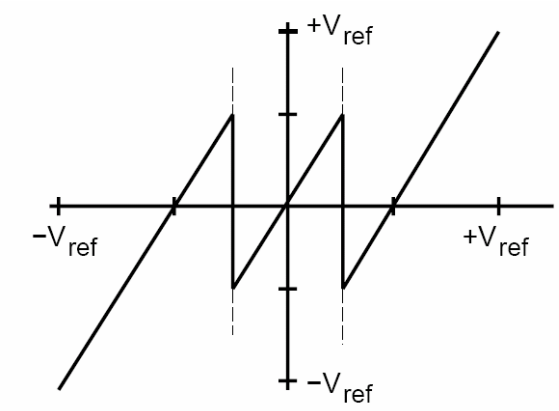
[Abo 1999] ([Lewis 1992])



- 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)

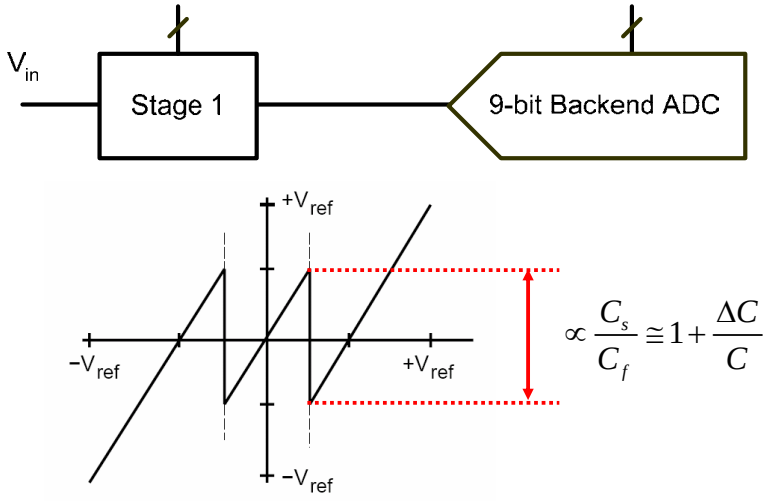
Residue Plot

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



[Abo 1999]

Stage 1 Matching Requirements



$$\propto \frac{C_s}{C_f} \cong 1 + \frac{\Delta C}{C}$$

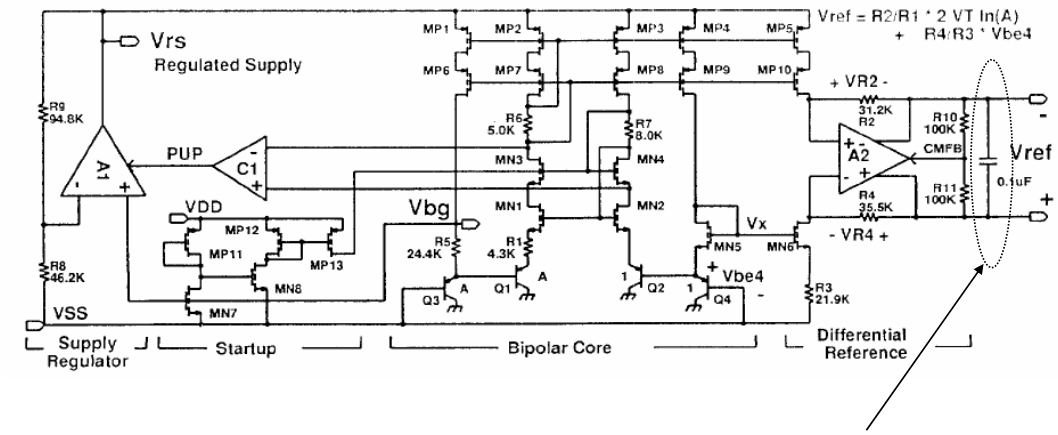
- Error in residue transition must be accurate to within a fraction of 9-bit backend LSB
  - Typically want $\Delta C/C$ ~0.1% or better

Capacitor Matching

- 0.1% "easily" achievable in current technologies
  - Even with metal sandwich caps, see e.g. [Verma 2006]
    - Beware of metal density related issues, "copper dishing"
  - For MIMCap matching data see e.g. [Diaz 2003]
- What if we needed much higher resolution than 10 bits?
  - Digital calibration
  - Multi-bit first stage
    - Each extra bit resolved in the first stage alleviates precision requirements on residue transition by 2x
    - For fixed capacitor matching, can show that each (effective) bit moved into the first stage
      - Improves DNL by 2x
      - Improves INL by sqrt(2)x
    - Multi-bit examples: [Singer 1996] [Kelly 2001] [Lee 2007]

Typical Reference Generator

[Brooks 1994]

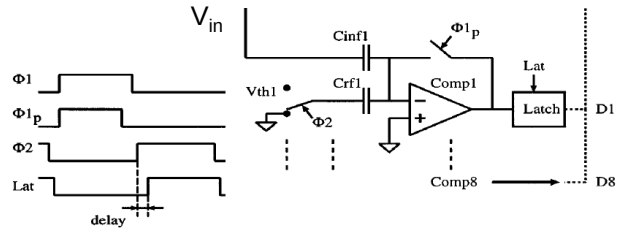


External decoupling caps provide dynamic currents
⇒ Low power reference buffer

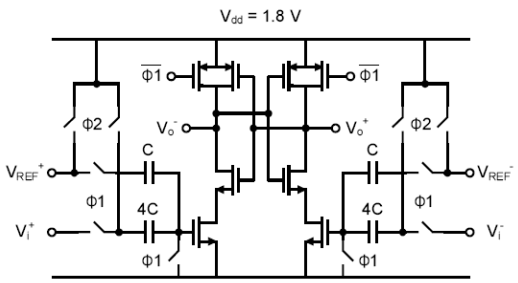
Comparators

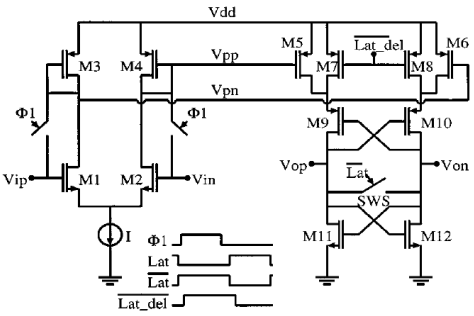
- Can tolerate large offsets and large noise with appropriate redundancy
- Consume negligible power in a good design
  - 50-100μW or less per comparator
- Lots of implementation options
  - Resistive/capacitive reference generation
  - Different pre-amp/latch topologies
  - …

Comparator Examples



[Chiu 2004]





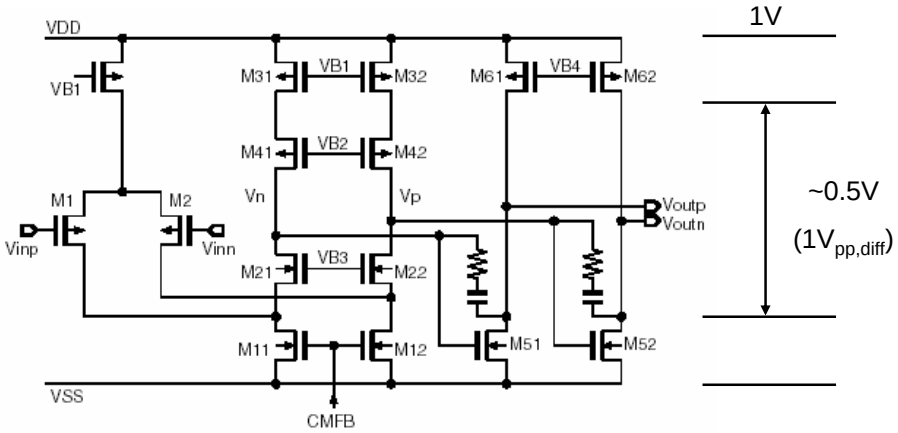
[Mehr 2000]

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 2007]

Two-Stage Folded Cascode OTA

[Ishii 2005]



- Works down to VDD=1V with reasonable output swing
- Gain ~ $(g_m r_o)^3$ ~ 10^3 = 60dB
- Use gain boosting to achieve larger gain

How Fast Can We Go? (1)

- Non-dominant pole in two-stage amplifier hard to move past $f_T/5$
- For 73 degrees phase margin (optimum for fast settling), loop crossover frequency is 1/3 of non-dominant pole frequency
- Settling linearly to 0.1% precision takes 7 loop time constants; typically budget ~10 time constants
- Ideally, we'd have 1/2 clock cycle to settle linearly, but there is some time needed for slewing and non-overlap clock timing
  - Assume 60% of half cycle is available for linear settling
- In summary

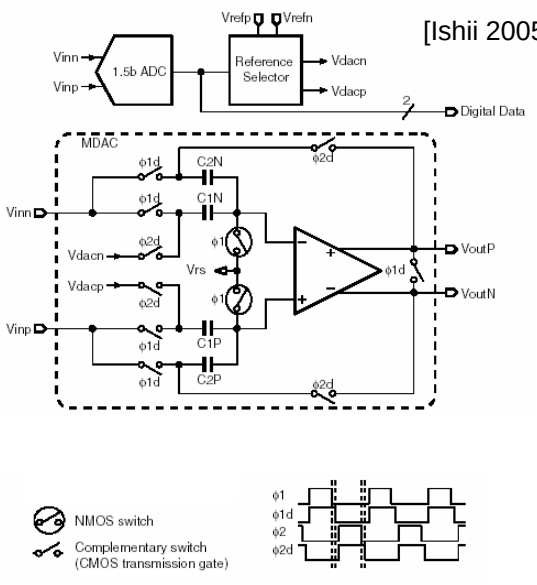
$$f_{CLK,max} = \frac{f_T}{5}\frac{1}{3}\cdot 2\pi \cdot \frac{1}{10}0.5\cdot 0.6 = \frac{f_T}{80}$$

How Fast Can We Go? (2)

Technology	NMOS f_T (at moderate V_{GS}-V_t ~150mV)	$f_{CLK,max} = f_T/80$
0.35um	10GHz	125 MHz
0.18um	30GHz	375 MHz
90nm	90GHz	1.125GHz (?)

- Sampling speeds of 200-300MHz are "easily" achievable in today's technologies
  - f_T is no longer a showstopper
  - Speed ultimately constrained by power, power efficiency and/or clock jitter

Switches



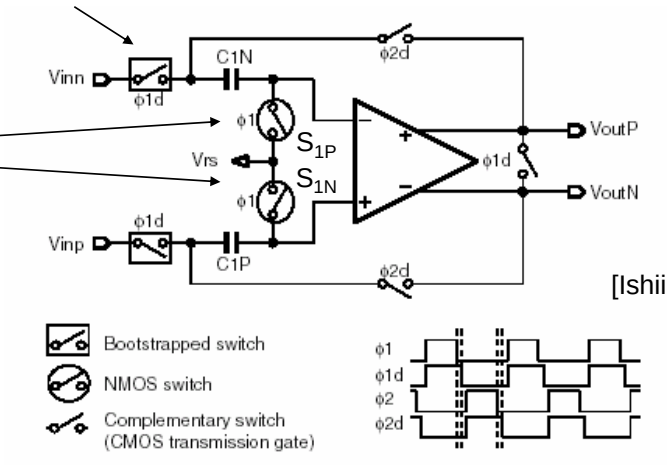
[Ishii 2005]

- 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

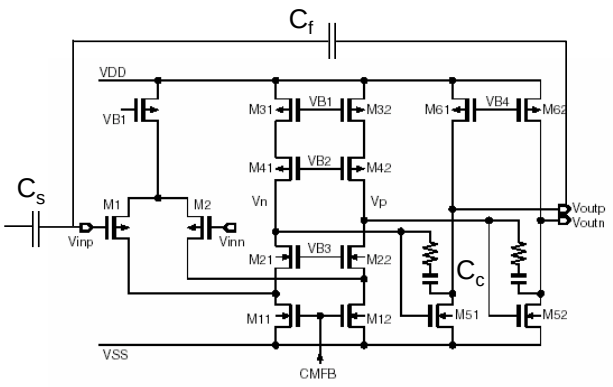
Need constant R_{ON} here to minimize signal dependent charge injection from S_{1N}, S_{1P}

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

$$\beta = \frac{C_f}{C_f + C_s + C_{gs1}} \qquad C_{Ltot} = C_L + (1-\beta)C_f + C_{parasitic}$$

Total Integrated OTA Noise (2)

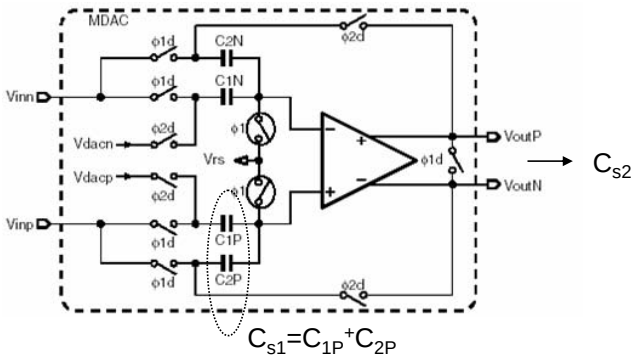
- Assuming γ=1, N_1=N_2=2

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

- 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



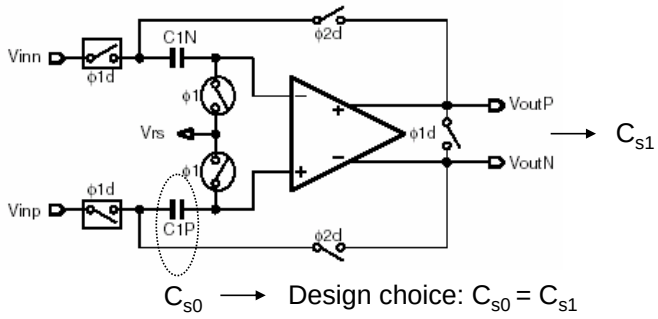
$$\beta = \frac{C_{s1}/2}{C_{s1}+C_{gs1}} \cong \frac{1}{3}$$

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

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

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

SHA Noise



C_{s0} → Design choice: $C_{s0} = C_{s1}$

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

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

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

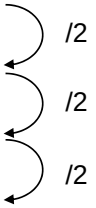
- Total input referred noise budget, assuming $V_{FS,diff}$=1V
  - $N_{thermal} = N_{quant}$=LSB^2/12=(1V/2^{10})^2/12=(280μVrms)^2
- Reasonable "first cut" partitioning of input referred noise
  - SHA → 1/2
  - Stage 1 → 1/4
  - All remaining stages → 1/4

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

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

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)

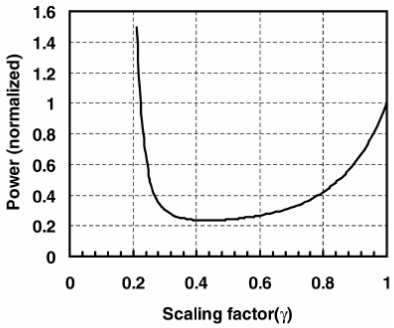


Fig.3. Power versus scaling factor of capacitors (10b ADC).

[Kawahito 2006]

(using single-stage OTA model)

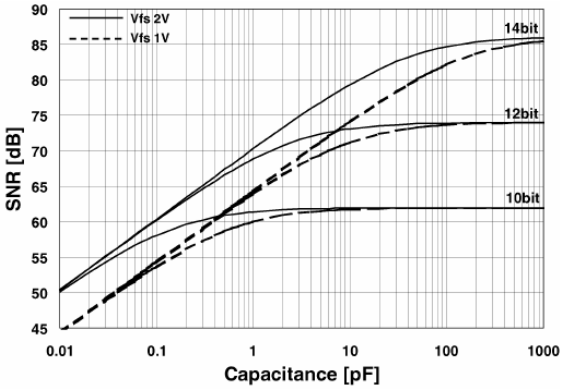


Fig. 9 SNR versus the sampling capacitor of the first stage of the MDAC (n=2, a=1, b=0.5, ζ=2, γ=0.5) .

Kawahito's Design Charts (2)

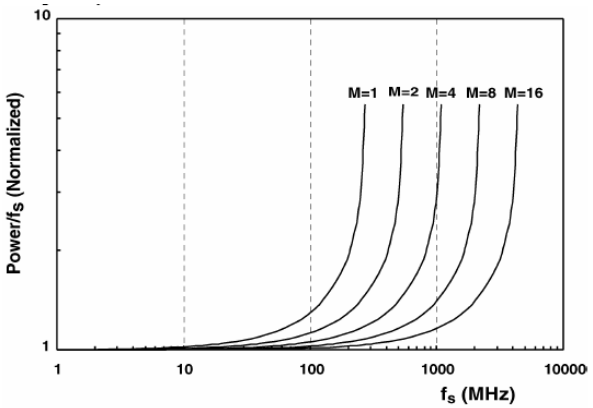


Fig. 13. Power/f_s versus f_s in parallel pipeline ADCs (# of channels (M) is 1, 2, 4, 8 and 16).

- Consider time-interleaving at high f_s

[Kawahito 2006]

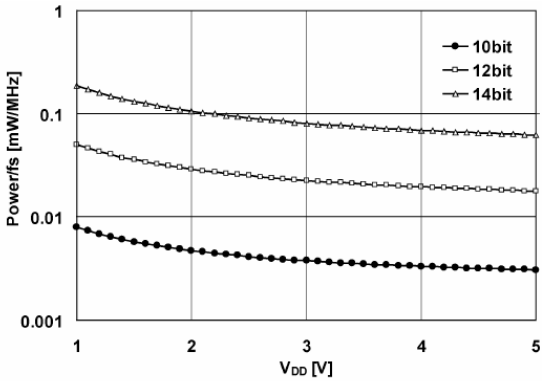
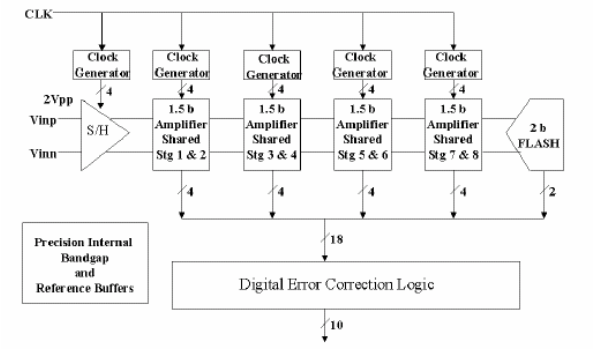


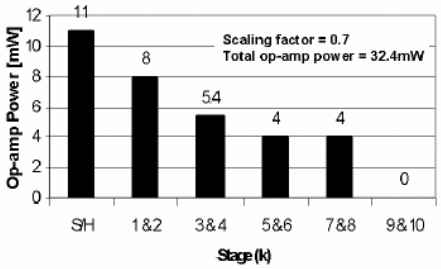
Fig. 14 Power normalized by sampling frequency of ideal pipeline ADCs versus V_{DD}.

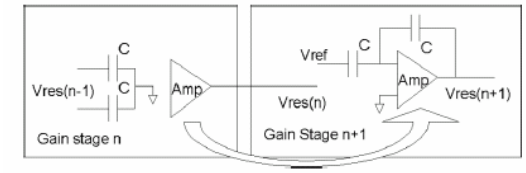
- In theory, plenty of room for power improvement…

Amplifier Sharing (1)



[Min 2003]

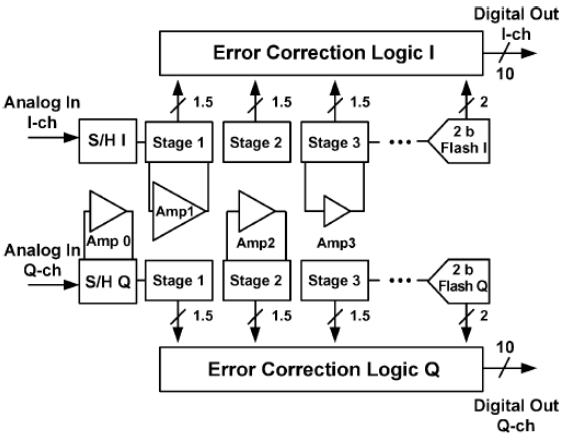




- Limited power savings because amplifiers have different specs

Amplifier Sharing (2)

[Kurose 2006]

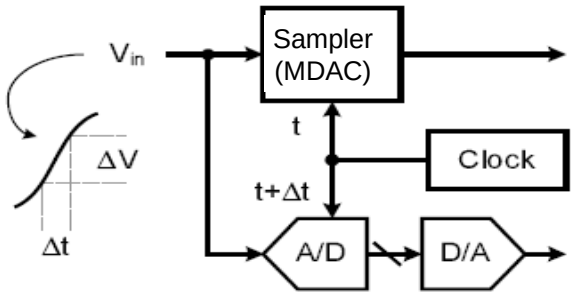


Technology	90-nm 1-P 7-M CMOS
Supply Voltage	1.2 V
Resolution	10 bit
Sampling Rate	200 MSPS
Full Scale	$0.8V_{p-p}$
DNL	+0.66/-0.61 LSB
INL	+0.90/-1.00 LSB
SFDR	66.5 dB
SNR	57.4 dB@Fin=9.9 MHz 55.6 dB@Fin=89.9 MHz
SNDR	54.4 dB@Fin=9.9 MHz 53.6 dB@Fin=89.9 MHz
ENOB	8.7 bit@Fin=9.9 MHz 8.6 bit@Fin=89.9 MHz
I/Q Isolation	>59 dB
Area	1.8 mm × 1.4 mm (2 ch)
Power	54.6 mW/ch

- Sharing of amplifiers is most efficiently done in a pair of converters that process I/Q signals

SHA-Less Architectures (1)

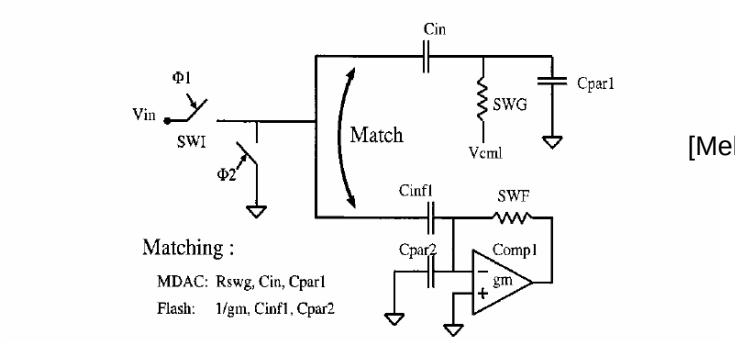
- Motivation
  - SHA can burn up to 1/3 of total ADC power
- Removing front-end SHA creates acquisition timing mismatch issue between first stage MDAC & Flash



[Chiu 2004]

SHA-Less Architectures (2)

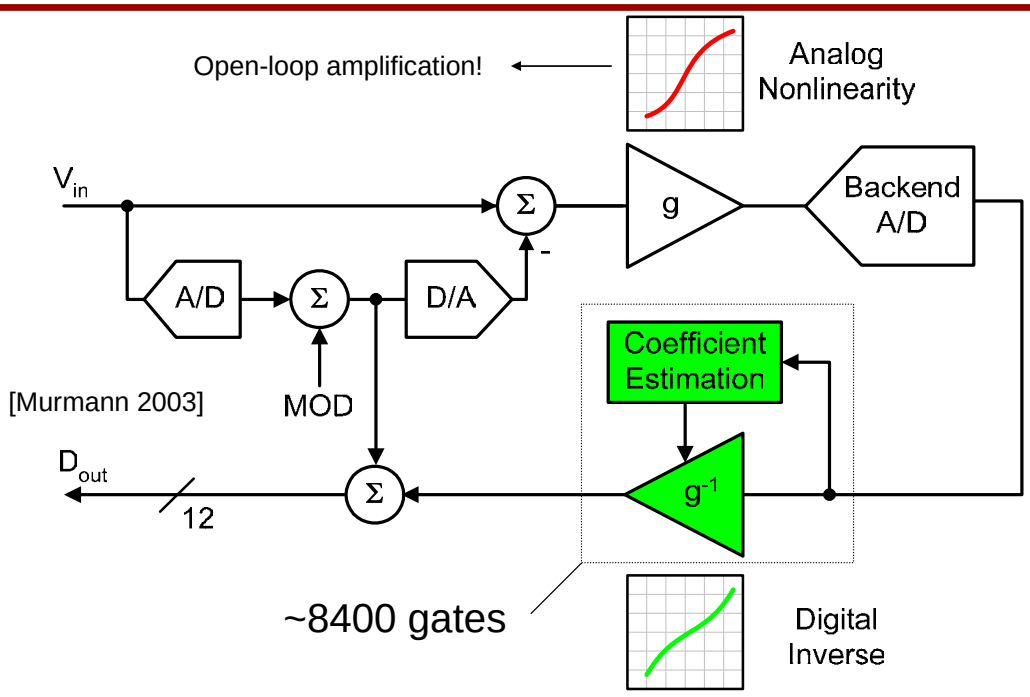
- Strategies
  - Use first stage with large redundancy; this can help absorb fairly large skew errors
  - Try to match sampling sub-ADC/MDAC networks
    - Bandwidth and clock timing



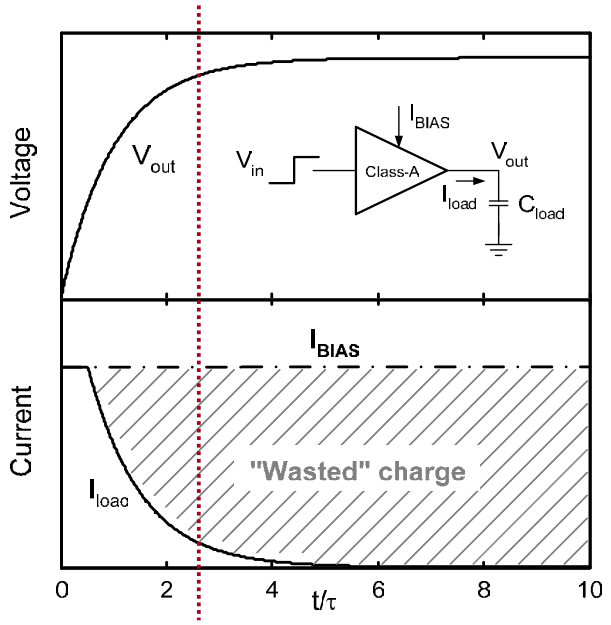
[Mehr 2000]

Fig. 5. Matching input networks for the MDAC and the flash comparators.

Research (1)



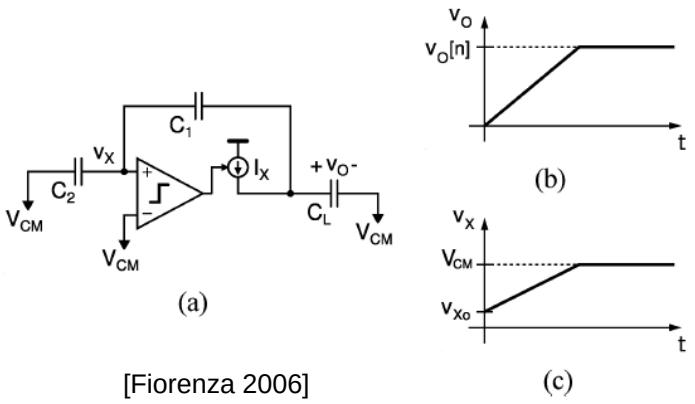
Research (2)



- Conventional designs settle to within small % error of final value (t_s/τ~10)
- Idea
  - Extend digital post-processing to correct for incomplete settling error
  - E.g. settle ~4x faster, or reduce power by ~4x

[Iroaga, 2007]

Research (3)



[Fiorenza 2006]

- Comparator-based switched-capacitor circuits
- Output slews to final value
  - Most efficient way to transfer charge

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.

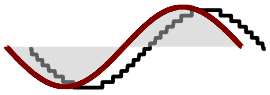
Selected References (4)

- 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, "EE214 Course Reader," Stanford University, Fall 2007, http://eeclass.stanford.edu/cgi-bin/handouts.cgi?cc=ee214
  - 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," to appear, VLSI Circuits Symposium, June 2008.

Lecture 14
Bit-at-a-Time ADCs
Time Interleaving

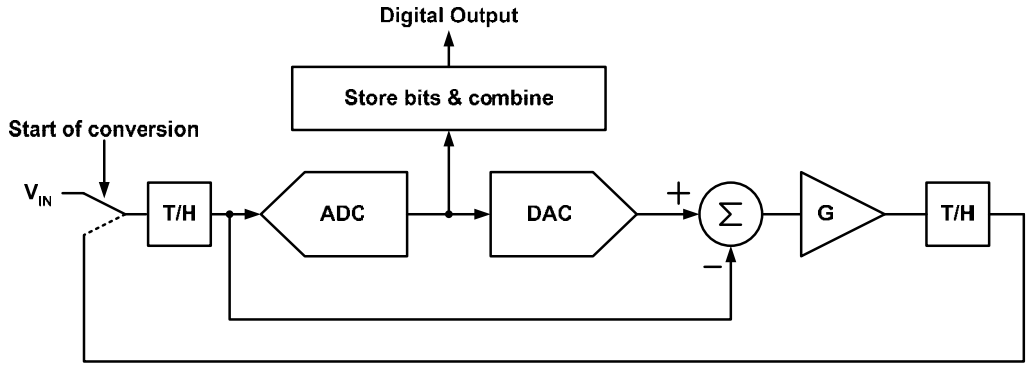


Boris Murmann
Stanford University
murmann@stanford.edu

Overview

- Bit-at-a-time ADCs
  - Cyclic ADCs
  - Successive approximation ADCs
- Time Interleaving
  - Use several Nyquist ADCs (any architecture) in parallel to increase conversion rate

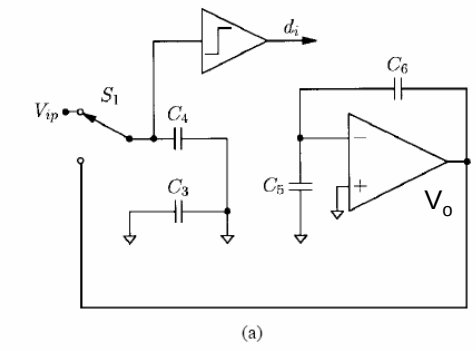
Cyclic ADC



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

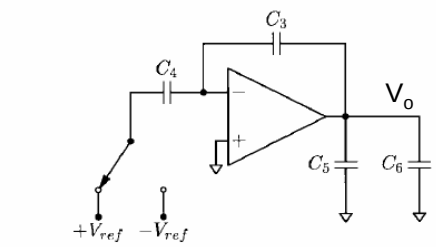
Implementation Example

ϕ1



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

ϕ2



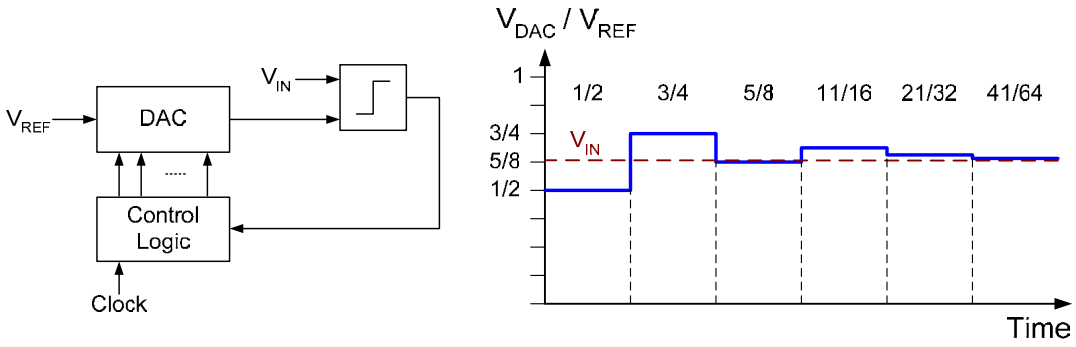
[Erdogan et al. JSSC 12/99]

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

Discussion

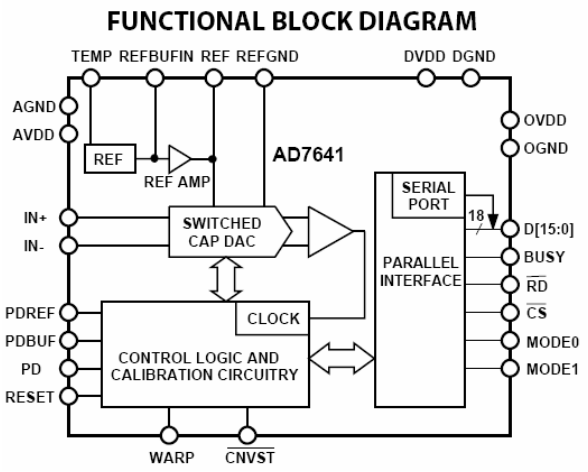
- Advantages
  - Area efficient
    - Typically only one or two switched capacitor stages plus comparator
  - Easy to calibrate
    - Need to measure only one coefficient (capacitor ratio)
- Disadvantages
  - Slow
    - Need many clock cycles for a single conversion
  - Sub-optimal power efficiency
    - 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)

High Performance Example

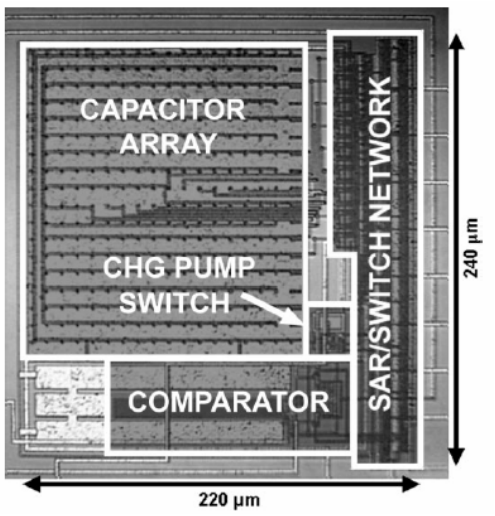


FEATURES

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



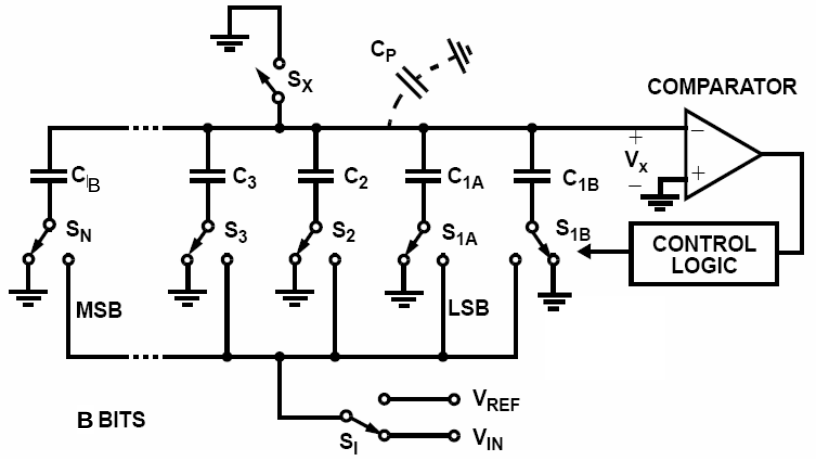
SUMMARY OF ADC PERFORMANCE

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.

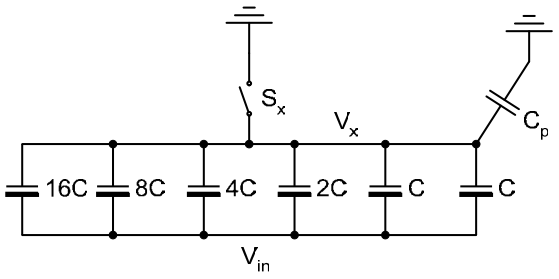
Implementation

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



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

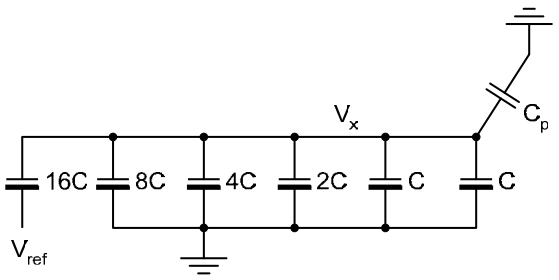
Sampling Phase (5-bit Example)



- Total charge at node V_x after opening S_x

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

Bit5 Test (MSB)

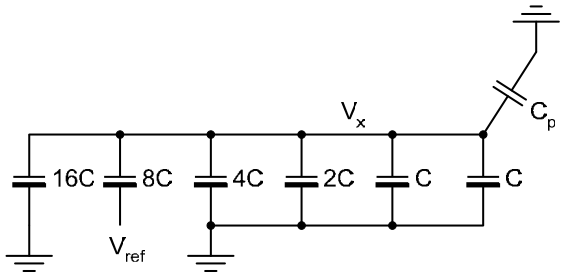


$$Q = -V_{in} \cdot C_{total} = (V_x - V_{ref}) \cdot 16C + V_x \cdot (16C + C_p)$$

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

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

Bit4 Test (Assuming bit5=0)



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

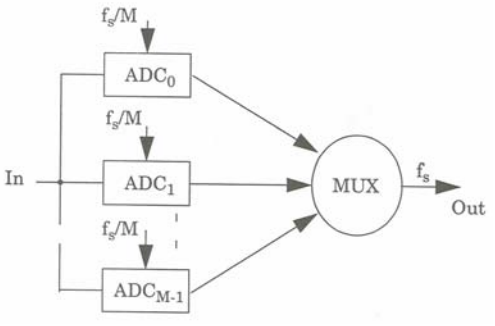
$$\therefore V_x = \left(\frac{1}{4}V_{ref} - V_{in}\right) \cdot \frac{C_{total}}{C_{total} + C_p}$$

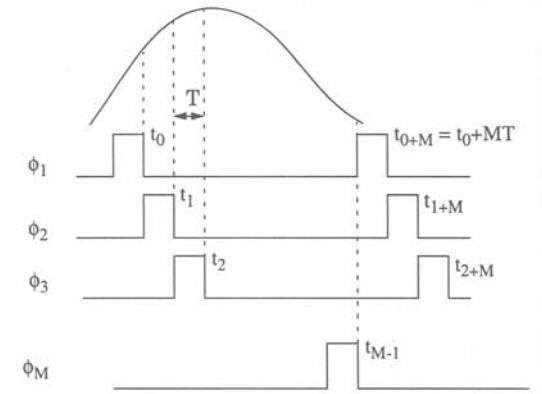
- $V_x<0 \Rightarrow V_{in}>0.25V_{ref} \Rightarrow$ Bit4=1
- $V_x>0 \Rightarrow V_{in}<0.25V_{ref} \Rightarrow$ Bit4=0

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]

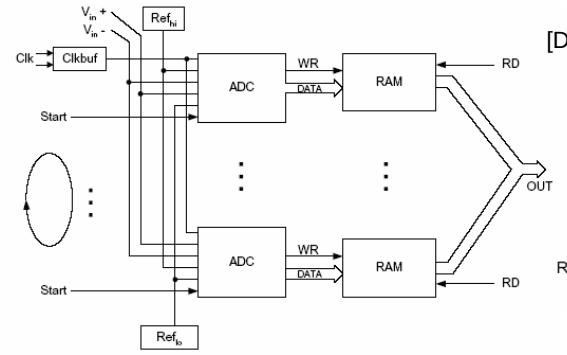
Time Interleaved ADCs



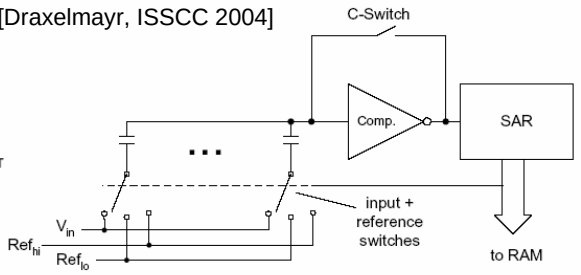


- Idea: Run M ADCs in parallel to obtain an aggregate throughput rate of $M \cdot f_s$
- Catch: Each ADC still needs an acquisition bandwidth that is commensurate with maximum input frequency

Example (1)

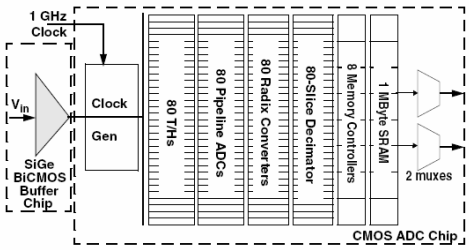


[Draxelmayr, ISSCC 2004]



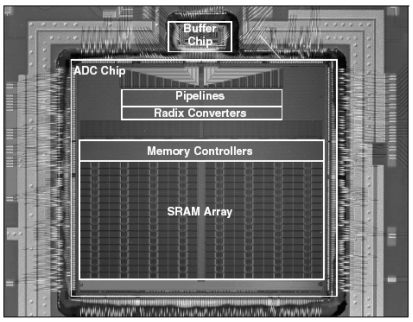
- Idea
  - Interleave several "slow" SAR ADCs to get high throughput while maintaining low complexity good power efficiency
- Array consists of eight 6-bit ADCs, each running at 75 MS/s
  - Aggregate throughput is 600MS/s, power=10mW in 90-nm CMOS technology

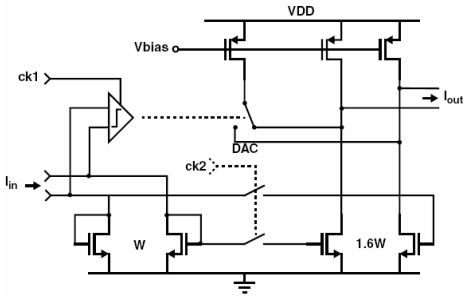
Example (2)



[Poulton, ISSCC 2003]

Sample Rate	20 GSa/s	
Resolution	8 bits	
INL Intrinsic With linearity correction	 ±1.7 LSBs ±0.4 LSBs	
DNL	±0.3 LSBs	
Bandwidth	6.6 GHz	
Accuracy @ 500 MHz input @ 6 GHz input	 6.5 effective bits 4.6 effective bits	
Jitter	0.7 ps rms	
Input Range	0.25 Vpk differential	
	Buffer Chip	**ADC Chip**
Input Capacitance	0.2 pF	4 pF
Power	1 W	9 W
Chip Size	1.2 x 2.6 mm	14 x 14 mm
Technology	40-GHz SiGe BiCMOS	0.18-mm CMOS
Transistors	1000	50M
Package	438-ball BGA	

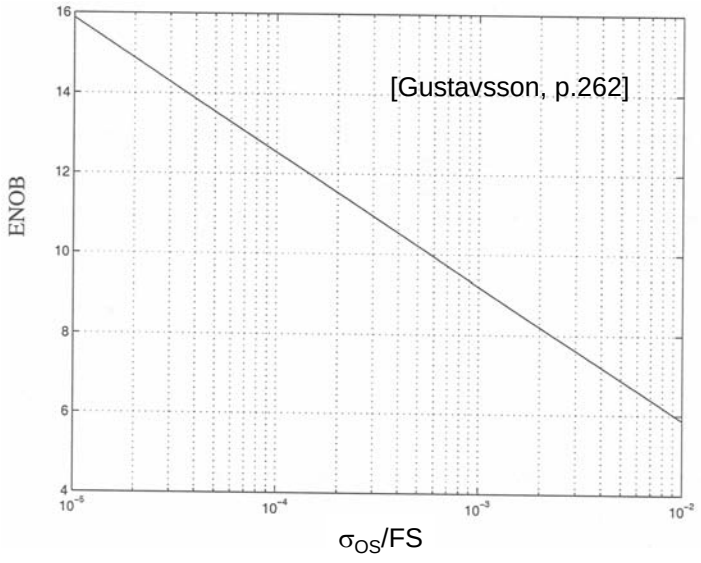




Issues with Time Interleaving

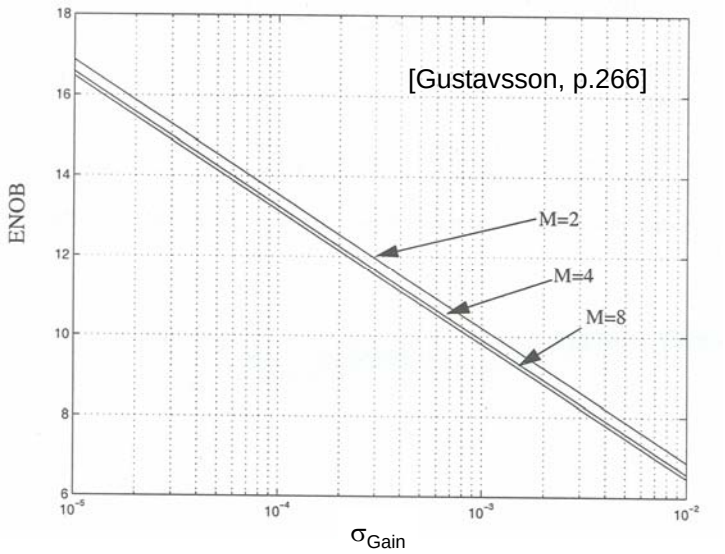
- Offset mismatch
  - Each channel will have a different offset
  - Output will contain a periodic error sequence that manifest itself as spurs in the output spectrum
- Gain mismatch
  - Channels may also have slightly different gain
  - Results in amplitude modulation
- Phase skew
  - Hard to guarantee precise phase relationship between individual channel clocks
  - Results in phase modulation (similar to aperture uncertainty)
- Solutions
  - "Careful design"
  - Analog or digital calibration
    - See e.g. [Jamal, JSSC 12/2002]

Impact of Offset Errors



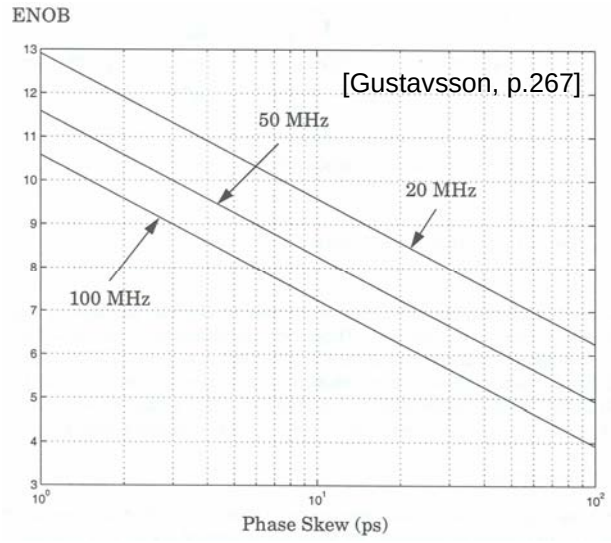
- E.g. FS=1V, σ_{OS}=1mV $\Rightarrow$ ENOB~9bits!

Impact of Gain Errors



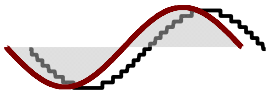
- E.g. σ_{Gain}=0.1% ⇒ ENOB~10bits

Impact of Phase Skew



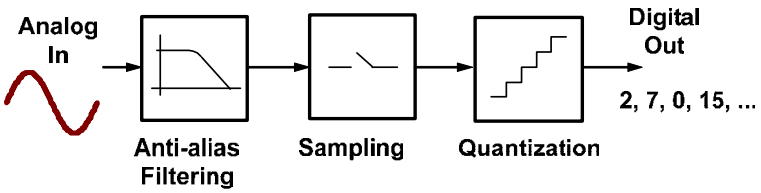
- Above chart is for M=4 channels
- E.g. f_{in}=100MHz, phase skew=3ps ⇒ ENOB~9bits!

Lecture 15
Oversampling A/D Conversion



Boris Murmann
Stanford University
murmann@stanford.edu

Recap

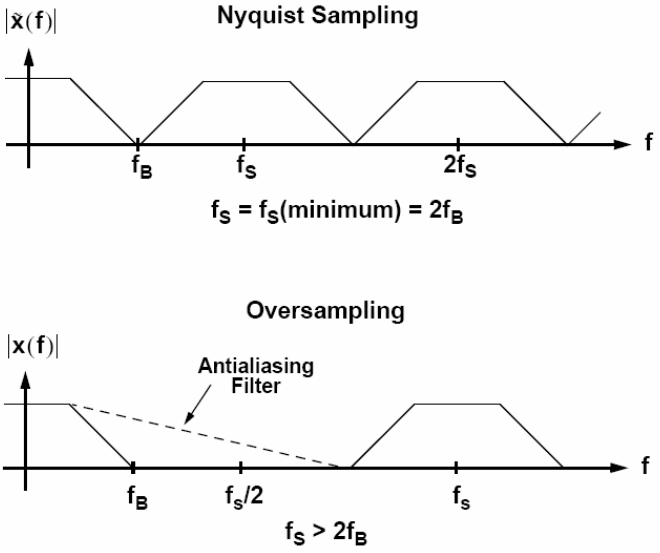


- Sampling theorem

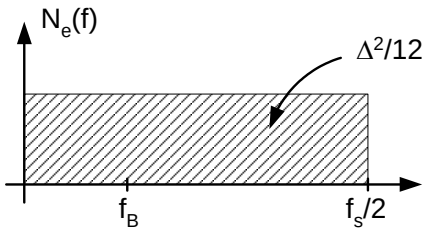
$$f_s > 2 f_{sig,max}$$

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

Anti-Alias Filtering

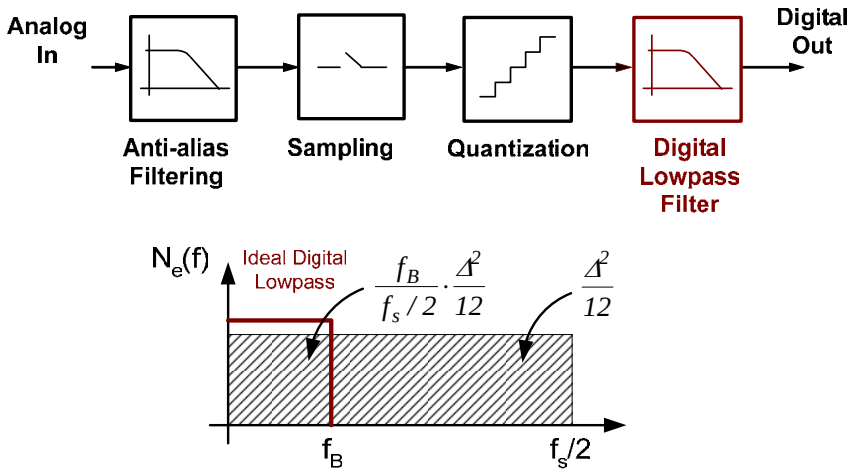


Quantization Noise



- Recall from lecture 2 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)



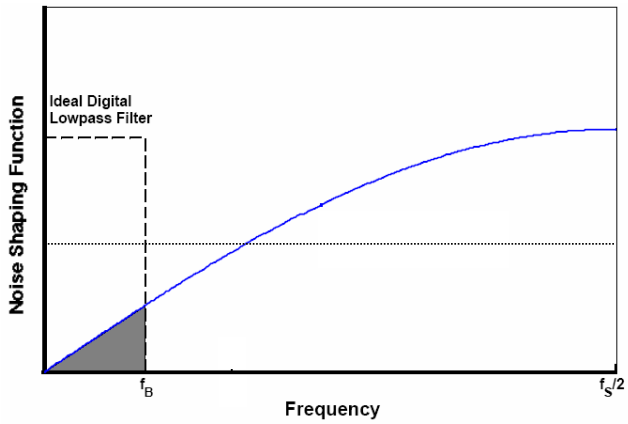
- Total quantization noise at digital output is reduced proportional to "oversampling ratio" $M=(f_s/2)/f_B$

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

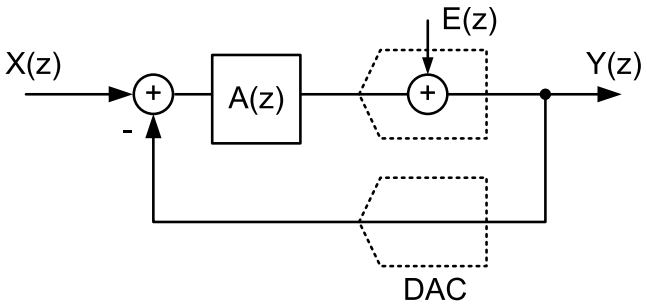
$$f_s \geq 2 \cdot f_B \cdot M = 2 \cdot 1MHz \cdot 2^{16}$$
$$\geq 131GHz$$

Noise Shaping



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

Noise Shaping Using Feedback (1)



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

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

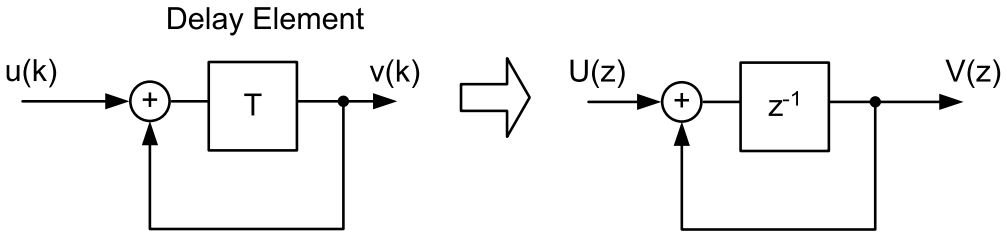
$$= E(z)\underbrace{H_E(z)}_{\substack{\textit{Noise}\\ \textit{Transfer}\\ \textit{Function}}} + X(z)\underbrace{H_X(z)}_{\substack{\textit{Signal}\\ \textit{Transfer}\\ \textit{Function}}}$$

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

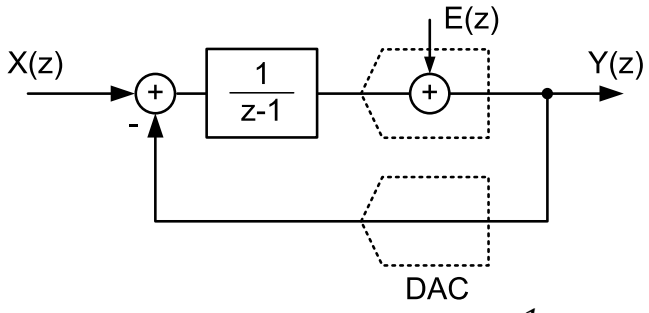


$$v(k) = u(k-1) + v(k-1) \qquad V(z) = z^{-1}U(z) + z^{-1}V(z)$$

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

- "Infinite gain" at DC (ω=0, z=1)

First Order Sigma-Delta 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

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

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

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

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

- The plot on slide 7 shows $|H_e(f)|$
  - "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 (shaded area on slide 7) and compare to full-scale signal

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

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

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

In-Band Quantization Noise (2)

- Assuming a full-scale sinusoidal signal, we have

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

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

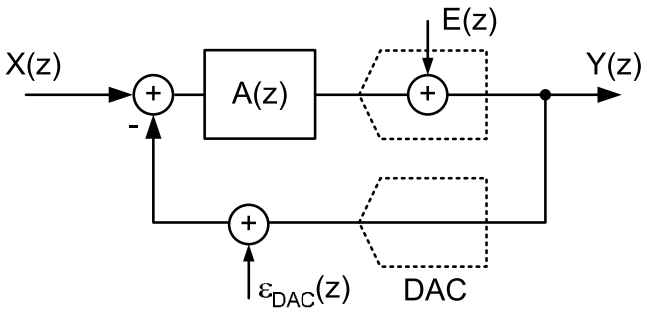
- Each 2x increase in M results in 8x SQNR improvement
  - 9dB (1.5bits) per octave oversampling

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



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

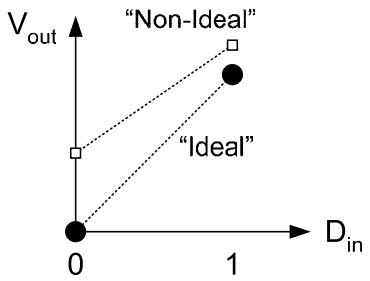
- DAC error is indistinguishable from signal
  - Means that DAC must be precise to within target resolution
- For the previous example, this means that we need an 8-bit DAC whose output levels have 16-bit precision…

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 quantizer

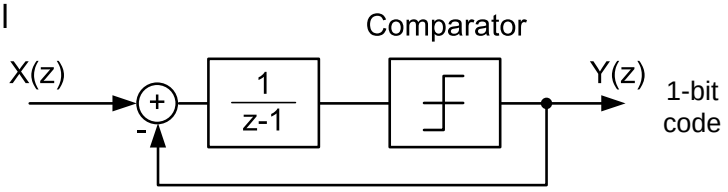
Single-Bit DAC

- 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



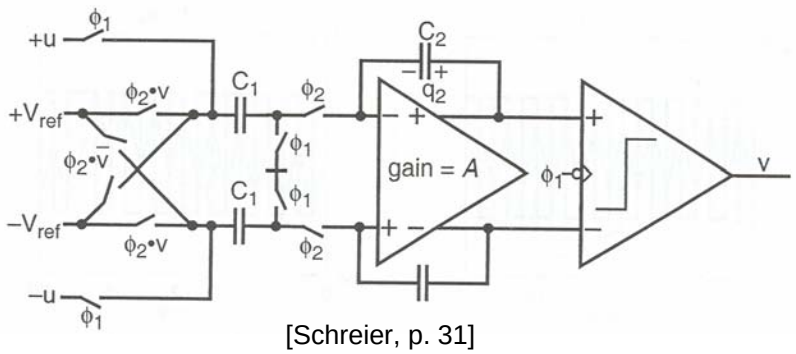
- Expected SQNR (from slide 14 with B=1)

$$SQNR \cong \frac{P_{sig}}{P_{qnoise}} = \frac{\frac{1}{2}\left(\frac{\Delta}{2}\right)^2}{\frac{\Delta^2}{12}\cdot\frac{\pi^2}{3}\frac{1}{M^3}} = \frac{9}{2\pi^2}\times M^3$$
$$= -3.4 + 30\log(M) \quad [dB]$$

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

Modulator with Single-Bit Quantizer (2)

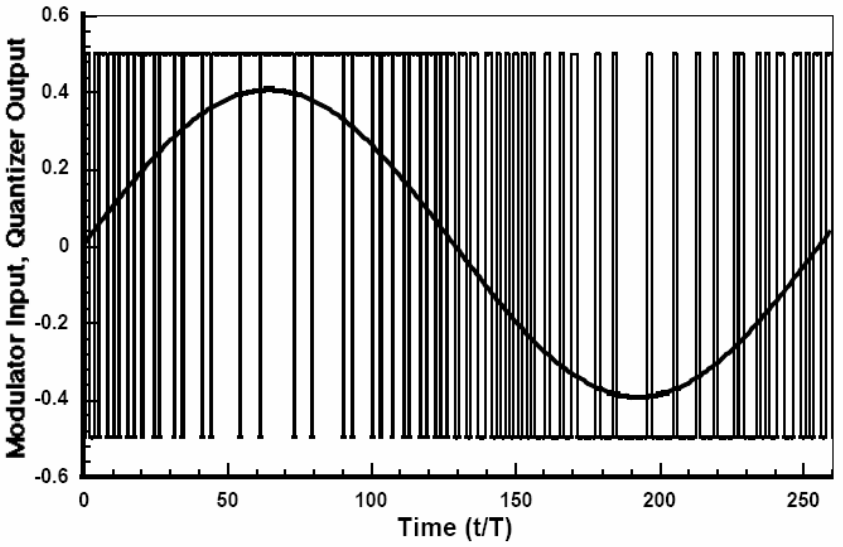
- Implementation example



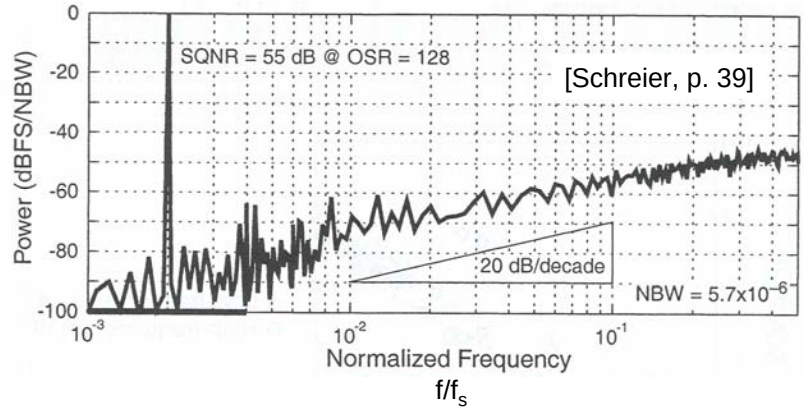
[Schreier, p. 31]

- 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

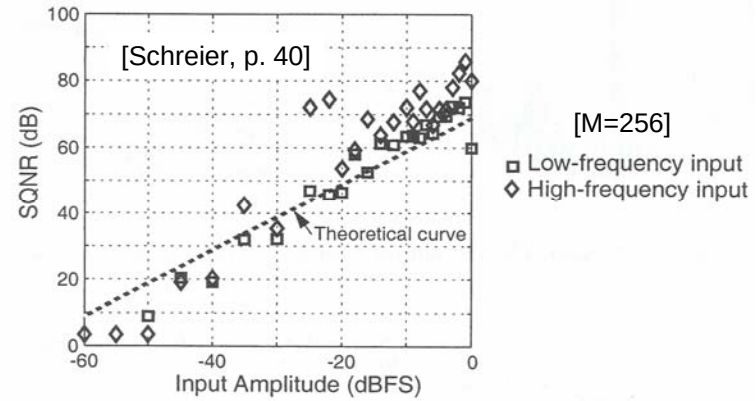


Spectrum



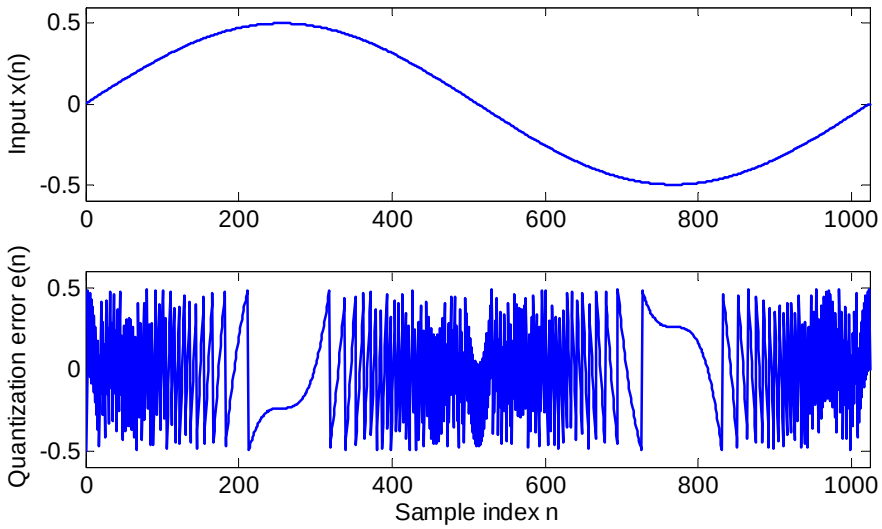
- Looks like there is some noise shaping, but SQNR=55dB is lower than the expected 60dB

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 (slide 4)

Quantization Error in 1st Order Modulator



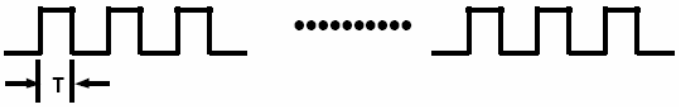
- A complicated, but deterministic function of the input

Tones

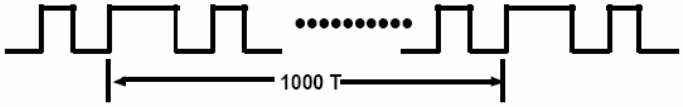
- Since the quantization error is correlated with the input, the shaped quantization noise contains spurious tones, some of which lie in the signal band
  - Spurs are visible on slide 22
- Linear model cannot predict these tones, but is still useful to gain insight into noise shaping process
- It is difficult to predict tonal behavior for arbitrary inputs
  - 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 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

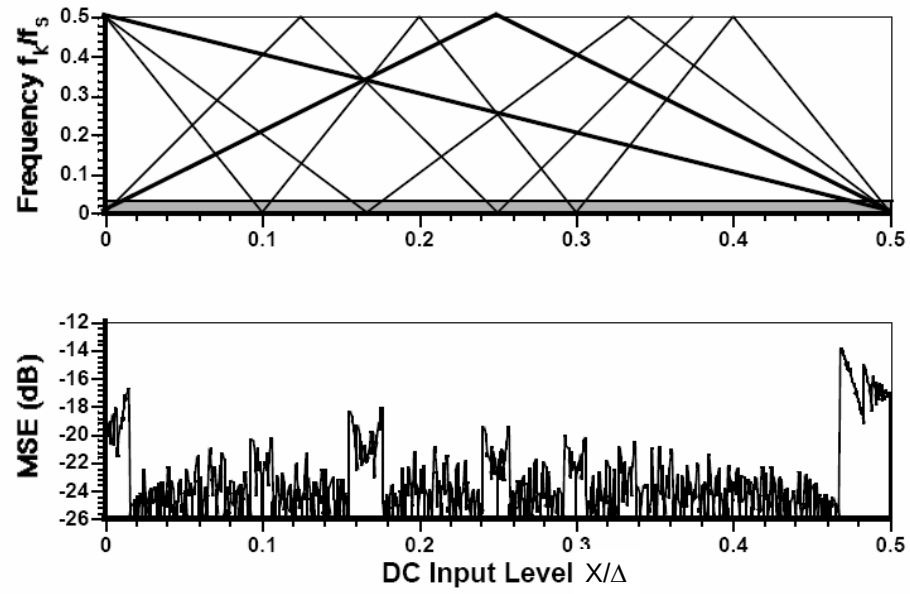
$$P_k = \left(\frac{\Delta \sin(\pi f_k T)}{\pi k} \right)^2$$

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

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
  - 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

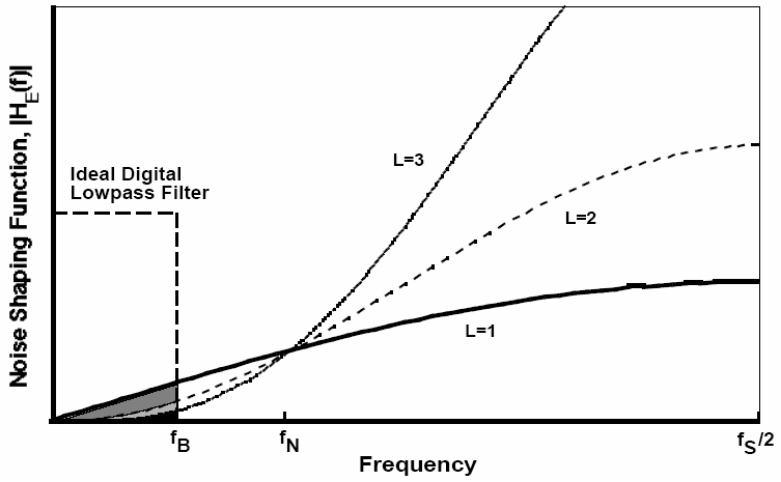
Higher Order Modulators

- 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

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

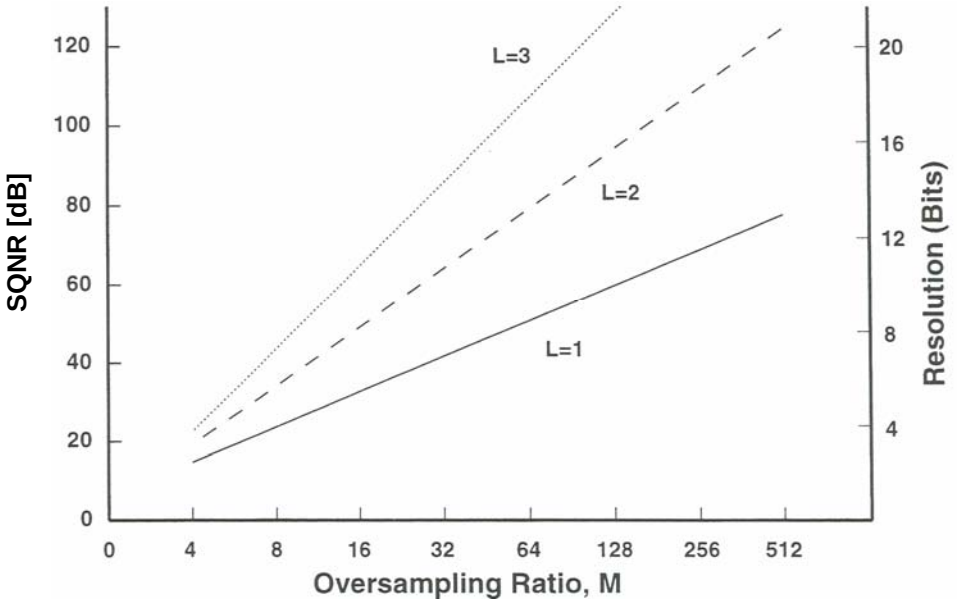
$$\cong \int_0^{f_B} \frac{\Delta^2}{12} \cdot \frac{2}{f_s} \cdot \left[2\pi \frac{f}{f_s} \right]^{2L} df$$

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

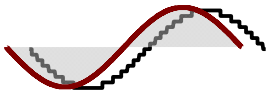
$$\cong \frac{\Delta^2}{12} \cdot \frac{\pi^{2L}}{2L+1} \left(\frac{1}{M} \right)^{2L+1}$$

- 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

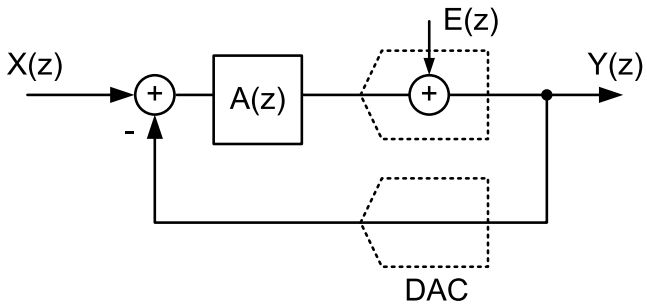


Lecture 16
Oversampling A/D Conversion
(Continued)



Boris Murmann
Stanford University
murmann@stanford.edu

Building a Second-Order Modulator (1)

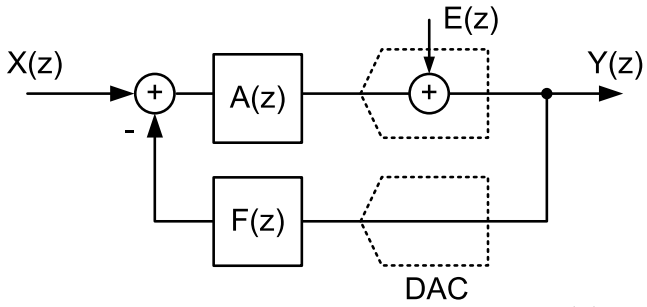


$$Y(z) = E(z)\frac{1}{1+A(z)} + X(z)\frac{A(z)}{1+A(z)}$$

- Want $\frac{1}{1+A(z)} = \left(1-z^{-1}\right)^2$ *and* $\frac{A(z)}{1+A(z)} = z^{-k}$

- Won't work without additional degree(s) of freedom…

Building a Second-Order Modulator (2)

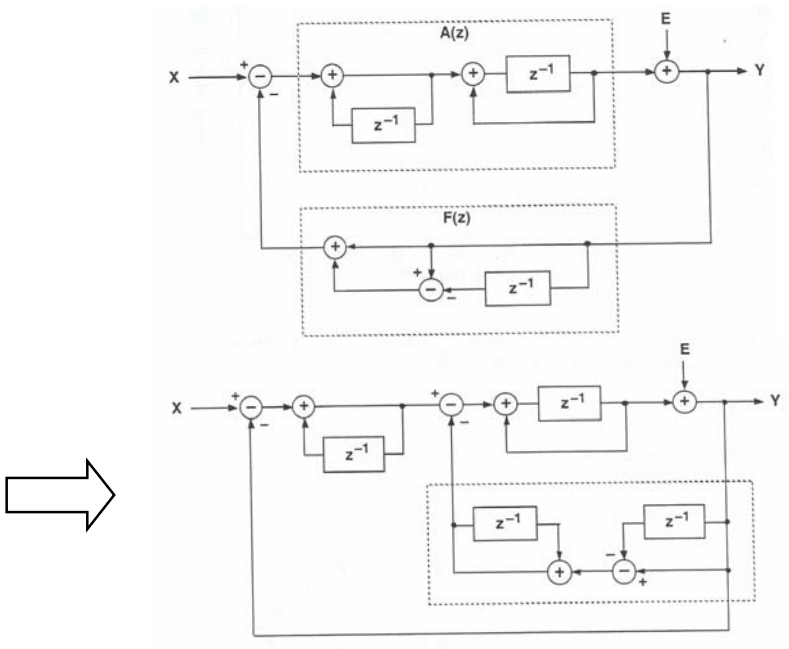


$$Y(z) = E(z)\frac{1}{1+A(z)F(z)} + X(z)\frac{A(z)}{1+A(z)F(z)}$$

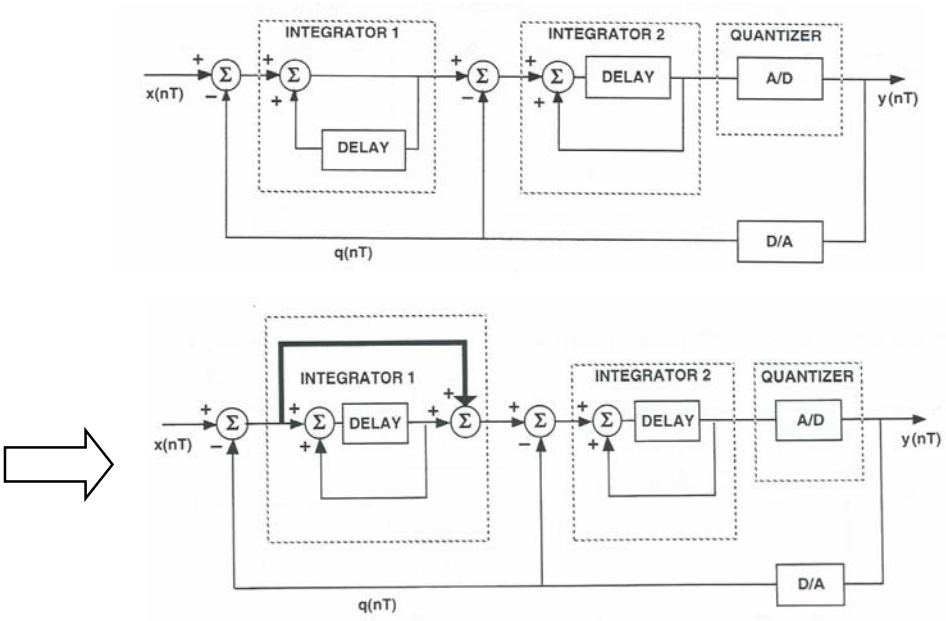
$$\frac{1}{1+A(z)F(z)} = (1-z^{-1})^2 \quad \text{and} \quad \frac{A(z)}{1+A(z)F(z)} = z^{-1}$$

$$\Rightarrow A(z) = \frac{z^{-1}}{(1-z^{-1})^2} \quad \text{and} \quad F(z) = 2 - z^{-1}$$

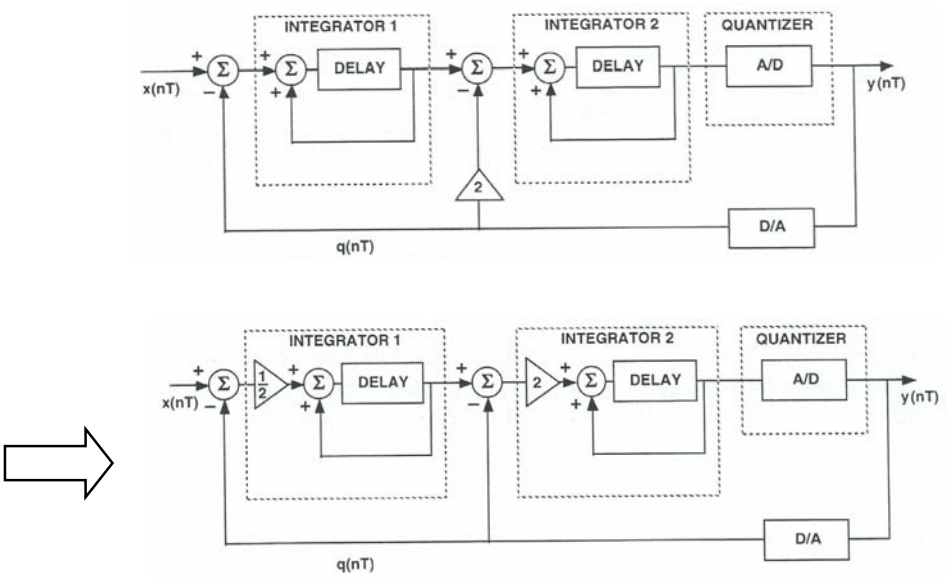
Building a Second-Order Modulator (3)



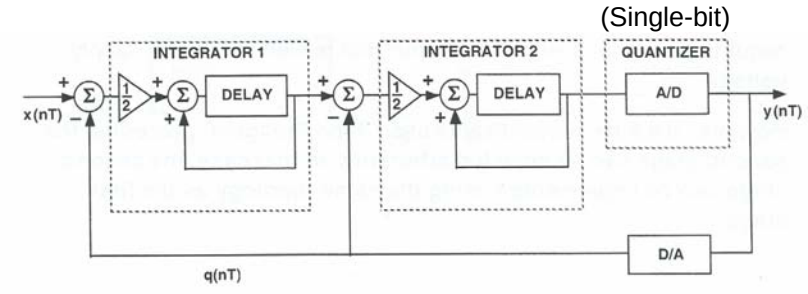
Building a Second-Order Modulator (4)



Building a Second-Order Modulator (5)



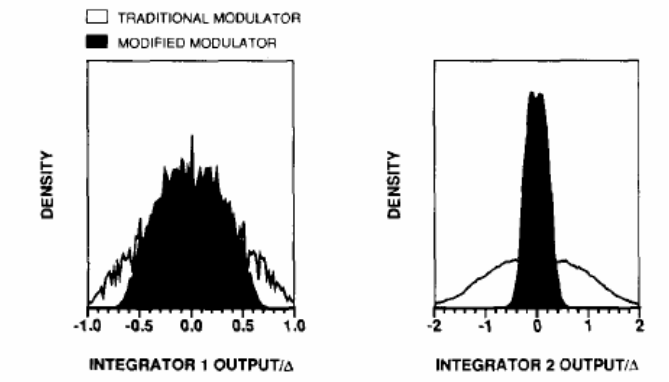
Boser-Wooley Modulator (1)



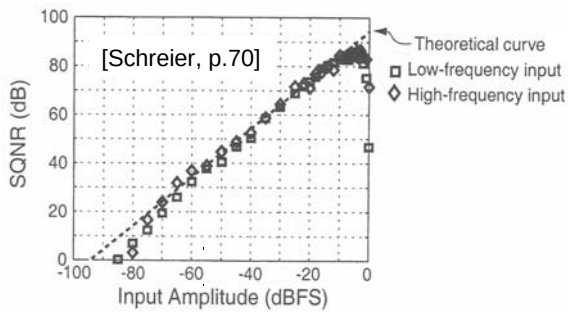
[Boser & Wooley, JSSC 12/1988]

- Two delaying integrators
  - Simplifies implementation
- Gain of second integrator scaled down to maximize useable swing at modulator input

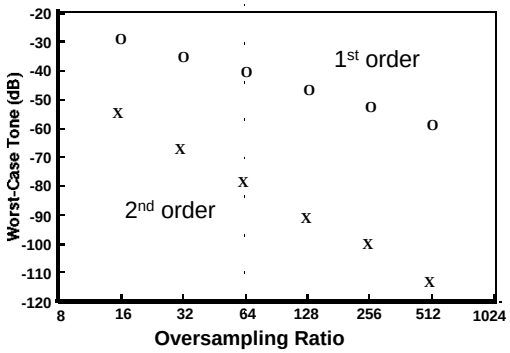
Boser-Wooley Modulator (2)



Performance of 2nd Order Modulator



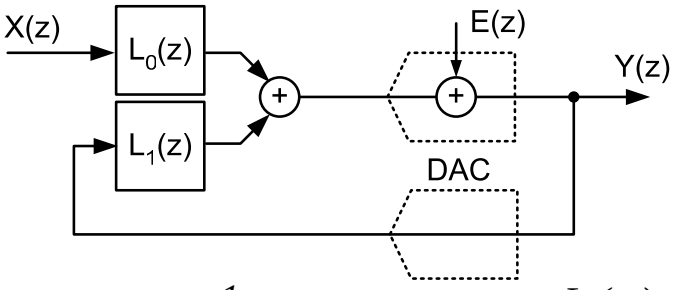
- Compared to first order modulator, SQNR is in "better" agreement with simple linear model



- Improved idle tone performance

Single Quantizer Modulators with Order >2

- Most general filter decomposition



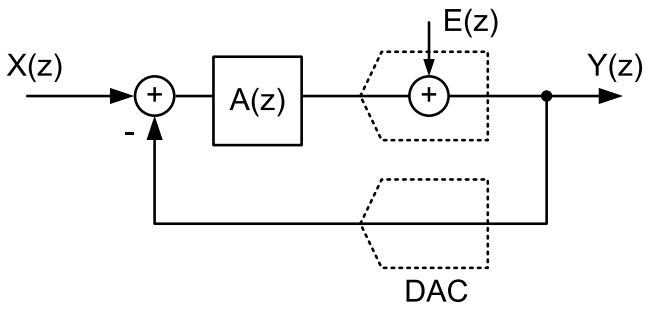
$$H_e(z) = \frac{1}{1 - L_1(z)} \qquad H_x(z) = \frac{L_0(z)}{1 - L_1(z)}$$

$$L_0(z) = \frac{H_x(z)}{H_e(z)} \qquad L_1(z) = 1 - \frac{1}{H_e(z)}$$

- $L_0(z)$ and $L_1(z)$ are usually implemented using shared hardware

Single Loop with High Order Filter

- Special case with L_0=A and L_1=-A



$$H_e(z) = \frac{1}{1 + A(z)} \qquad H_x(z) = \frac{A(z)}{1 + A(z)} \cong 1 \quad in\ band\ of\ interest$$

Stability

- Primarily determined by characteristics of $H_e(z)$
- 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
  - For "reasonable", slow varying inputs of magnitude $<0.8 \cdot \Delta/2$, integrator outputs are "likely" to stay within bounds
- To date, no exact stability criteria for higher order modulators have been found
  - Lee's criterion for single bit, high order modulators states that the modulator is "likely" to be stable if $\max[H_e(\omega)]<1.5$
- In practice, designers rely on a combination of stability analysis using the linear model (!) and simulations of the nonlinear model

Typical Design Procedure

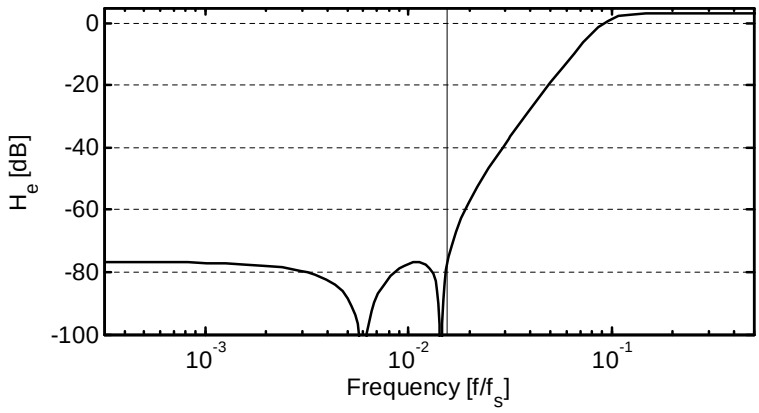
- "Cookbook design"
  - See e.g *Delta-Sigma Data Converters*, by Norsworthy, Schreier & Temes, Sections 4.4 and 5.6
  - Choose order based on desired SQNR and M
  - Design NTF using filter approximations (e.g. Chebyshev)
    - Make sure to obey Lee's criterion
  - Determine loop-filter transfer function 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
```

Possible Realization

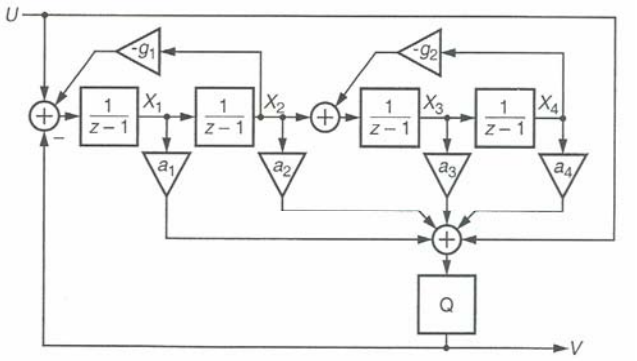


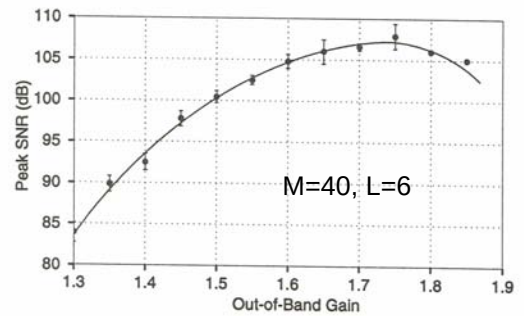
Figure 4.21: Chain of integrators with feedforward summation and local resonator feedbacks (modified CIFF structure).

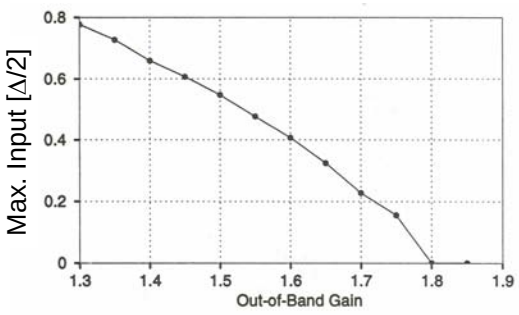
[Schreier, p. 123]

- Can show (with a little algebra) that coefficients of A(z) map directly into values for $a_1 \ldots a_4$, g_1 and g_2 in above realization

The Cost of Stability

[Norsworthy, pp.156]





- Higher out of band gain means higher attenuation in the signal band and hence better SQNR
  - Unfortunately modulator becomes "less stable"

Achievable SQNR

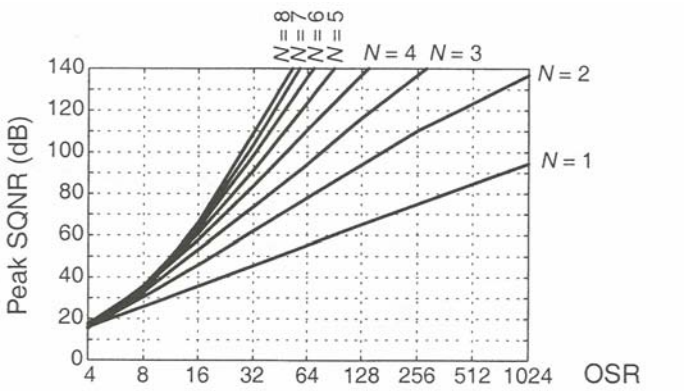


Figure 4.14: Empirical SQNR limit for 1-bit modulators of order N.

- Diminishing return for order greater 5-6

Commercial Example

ANALOG DEVICES

Single-Supply 16-Bit ΣΔ Stereo ADC

AD1877*

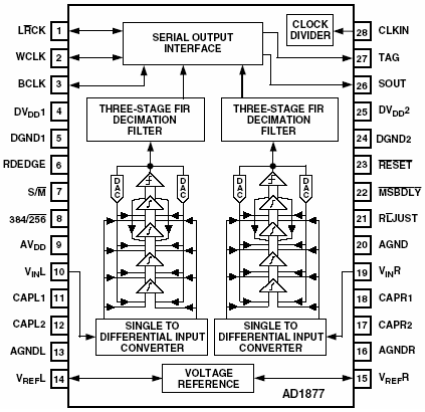
FEATURES

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

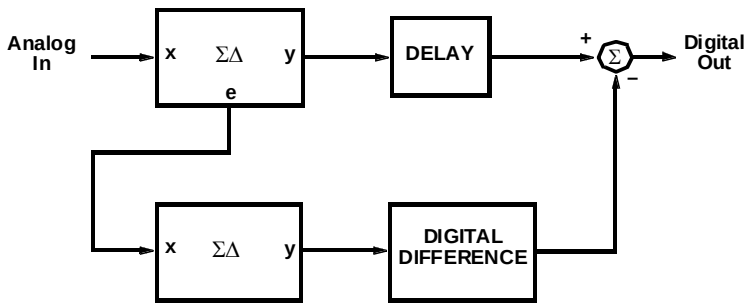
APPLICATIONS

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

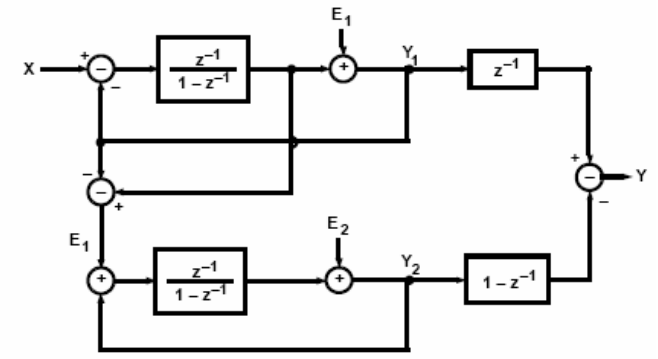


Cascaded Modulators



- 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



$$Y_1(z) = z^{-1}X(z) + (1 - z^{-1})E_1(z)$$

$$Y_2(z) = z^{-1}E_1(z) + (1 - z^{-1})E_2(z)$$

$$Y(z) = z^{-1}Y_1(z) - (1 - z^{-1})Y_2(z)$$

$$= z^{-2}X(z) + z^{-1}(1 - z^{-1})E_1(z) - z^{-1}(1 - z^{-1})E_1(z) - (1 - z^{-1})^2E_2(z)$$

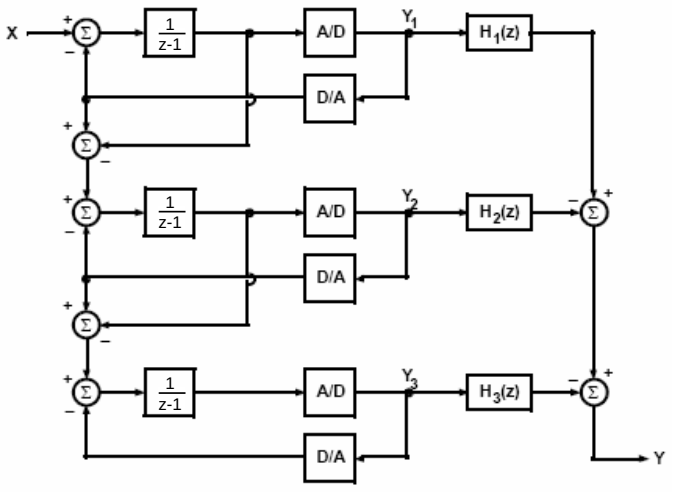
$$Y(z) = z^{-2}X(z) - (1 - z^{-1})^2E_2(z)$$

- Second order noise shaping using two first order loops!

Properties

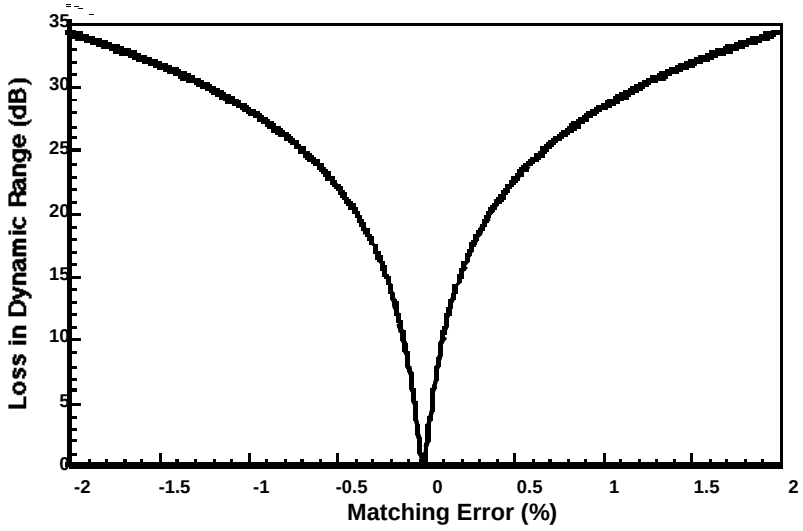
- Order of overall noise shaping is equal to sum of modulator orders
- No stability issues
- Improved idle tone performance
  - Input of second stage is "noise like"
  - Remaining quantization error from second stage is very close to white noise
- Cancellation of first stage quantization noise depends 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)

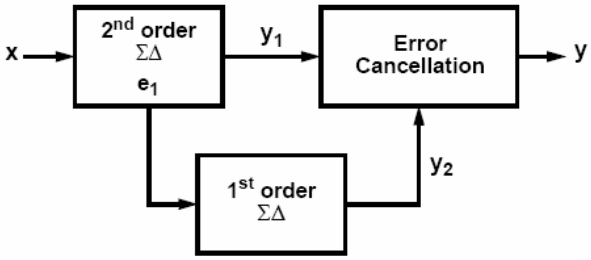


Mismatch Sensitivity

Sensitivity of 1-1-1 cascade to matching between the analog and digital "gains"



2-1 Cascade



$$Y_1(z) = z^{-2}X(z) + (1 - z^{-1})^2 E_1(z)$$

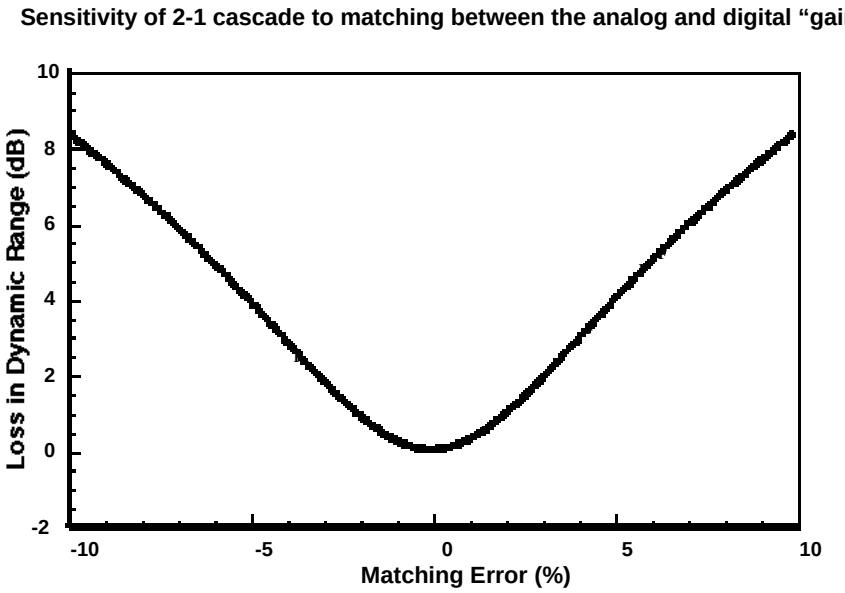
$$Y_2(z) = z^{-1}E_1(z) + (1 - z^{-1})E_2(z)$$

$$Y(z) = z^{-1}Y_1(z) - (1 - z^{-1})^2 Y_2(z)$$

$$= z^{-3}X(z) + z^{-1}(1 - z^{-1})^2 E_1(z) - z^{-1}(1 - z^{-1})^2 E_1(z) - (1 - z^{-1})^3 E_2(z)$$

$$Y(z) = z^{-3}X(z) - (1 - z^{-1})^3 E_2(z)$$

Mismatch Sensitivity

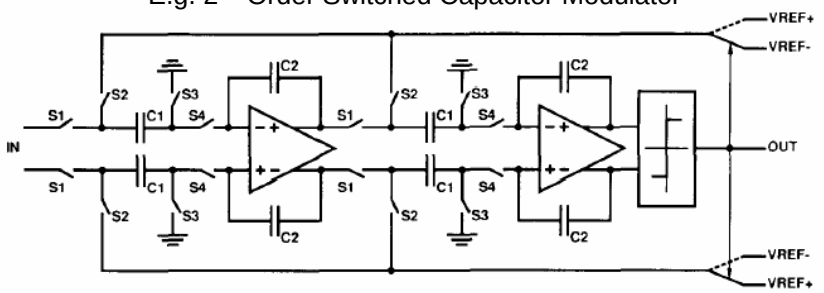


Circuit Level Considerations

- Electronic noise
- Finite OTA gain
  - Integrator leak
  - Dead zones
  - Nonlinearity
- OTA dynamic settling error, nonlinearity due to 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

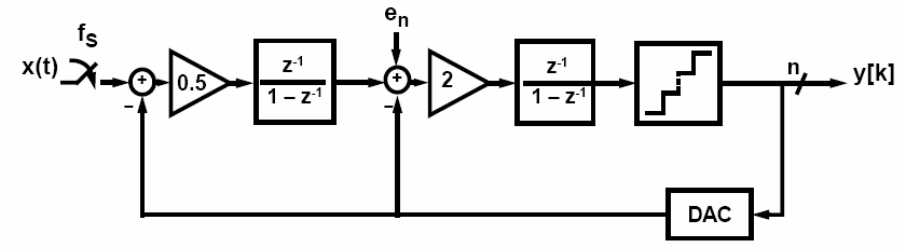
Electronic Noise

E.g. 2nd Order Switched Capacitor Modulator



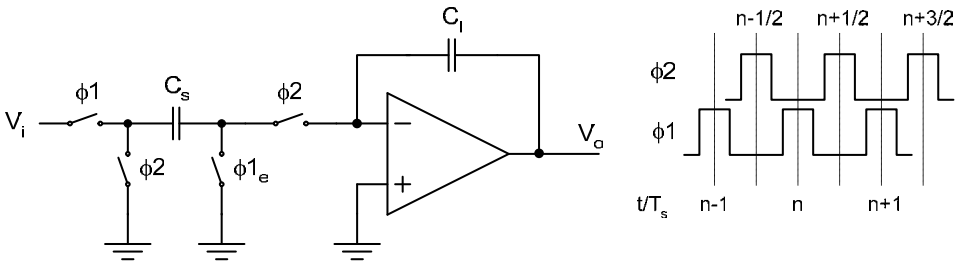
- Noise from 1st integrator is added directly to the input
  - Digital filter will reduce this noise by oversampling ratio
- Noise from 2nd integrator is first-order noise shaped!
  - Digital filter will remove most of this noise
- Especially for high oversampling ratios, only the first one or two integrators 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	…	…

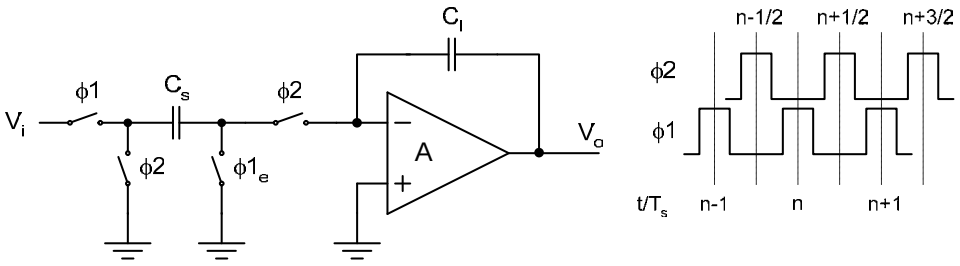
Integrator Analysis (1)

- 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

Finite Gain (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	…	…

Finite Gain (2)

- Again, assuming that V_o is sampled during $\phi 1$, we have

$$C_I V_o(z)\left[1+\frac{1}{A}\right] = z^{-1} C_I V_o(z)\left[1+\frac{1}{A}\right] + z^{-1} C_s V_i(z) - \frac{C_s}{A} V_o(z)$$

$$\therefore \frac{V_o(z)}{V_i(z)} \cong \frac{C_s}{C_I} \frac{z^{-1}\left(1-\frac{1}{A}\left[1+\frac{C_s}{C_I}\right]\right)}{1-\left(1-\frac{1}{A}\frac{C_s}{C_I}\right)z^{-1}} = \frac{g \cdot z^{-1}}{1-[1-\alpha]\cdot z^{-1}}$$

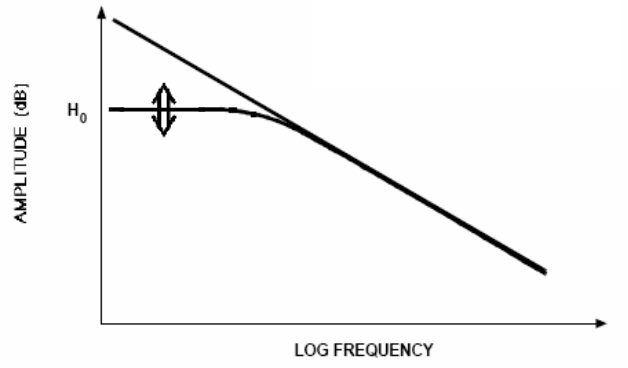
$$V_o(z) = [1-\alpha]\cdot z^{-1} V_o(z) + g \cdot z^{-1} V_i(z)$$

- Finite gain results in "leaky integrator"
  - Some fraction of previous output is lost in new cycle

Frequency Domain View

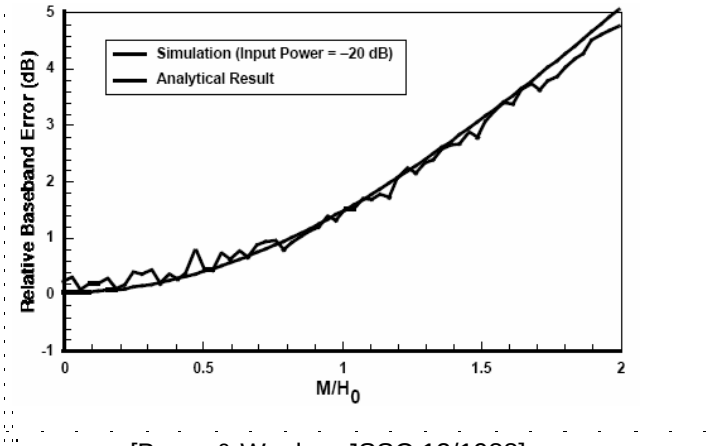
- Limited gain at low frequencies ($\omega \rightarrow 0$, $z \rightarrow 1$)

$$H_0 = H(z)\Big|_{z=1} = \frac{g}{1-[1-\alpha]} = \frac{g}{\alpha} \propto A$$



- But noise shaping relies on high integrator gain at low frequencies…

Required DC Gain



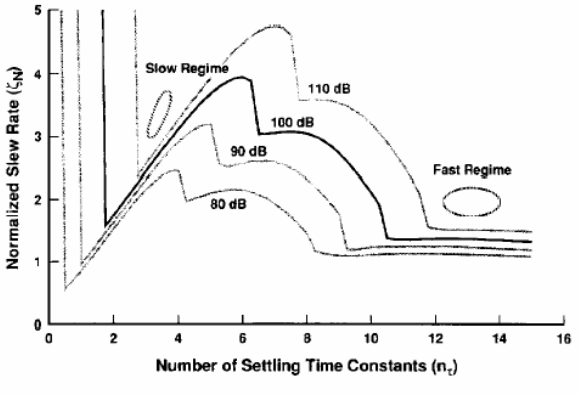
[Boser & Wooley, JSSC 12/1988]

- Good practice to make OTA gain at least a few times larger than oversampling ratio

Settling

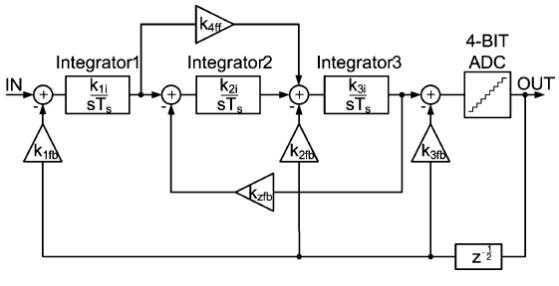
No Slewing, incomplete settling

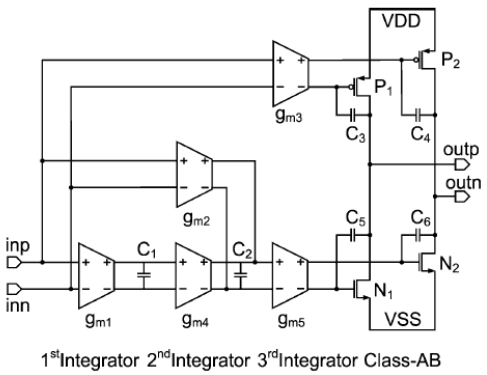
Slewing, amplifier settles



[Williams & Wooley, JSSC 3/1994]

State-of-the-art CT Delta-Sigma Modulator





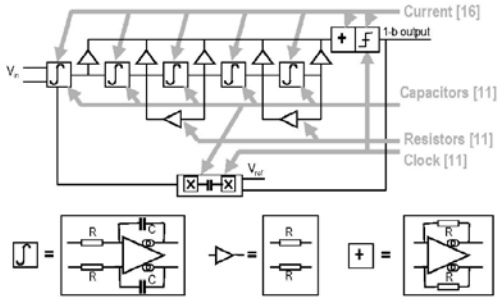
1st Integrator 2nd Integrator 3rd Integrator Class-AB

[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 sigma delta modulators

Multi-Mode Modulator

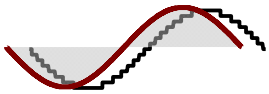


Process	1P 6M standard 90nm			
Supply voltage	1.1V – 1.3V (±3dB DR performance deviation)			
ΔΣ modulator	5th-order CT, feedforward, 1-b with SC DAC			
Input voltage range	0.45V_{rms}, differential			
Modes	121	GSM	BT	WLAN
Sampling rate	13MHz - 400MHz	26MHz	200MHz	400MHz
Signal bandwidth	100kHz - 10MHz	200kHz	1MHz	10MHz
Dynamic range	52dB - 82dB	82dB	75dB	62dB
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

Lecture 17
Decimation Filters
Oversampling D/A Conversion

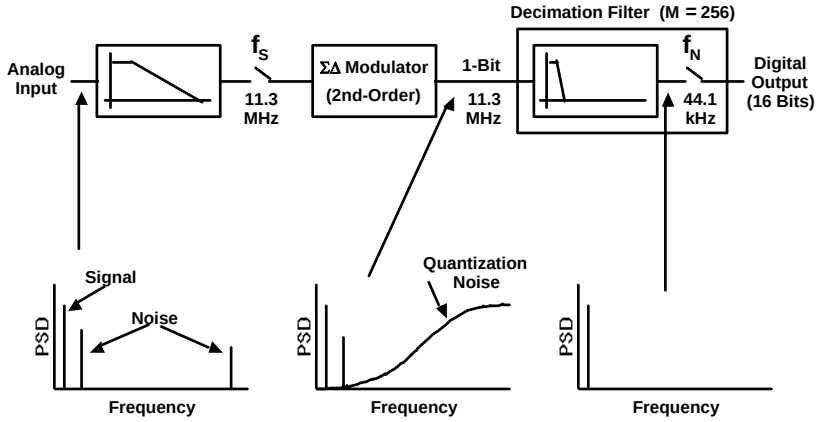


Boris Murmann
Stanford University
murmann@stanford.edu

Decimation Filters

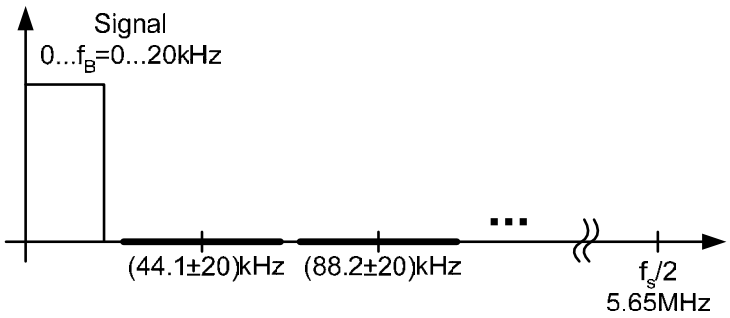
- References
  - J. Candy, "Decimation for Sigma-Delta Modulation," *IEEE Trans. Communications*, pp. 72-76, Jan. 1986.
  - Chapters 1 and 13 of *Delta-Sigma Data Converters*, by Norsworthy, Schreier, Temes.
  - B.P. Brandt and B.A. Wooley, "A low-power, area-efficient digital filter for decimation and 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

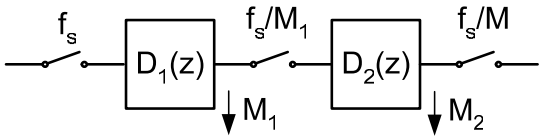
Filter Requirements



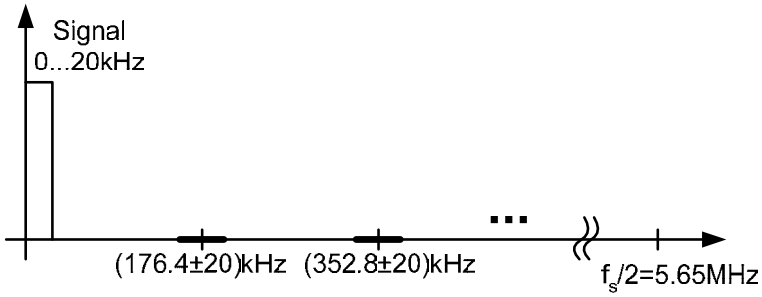
- Pass band 0…20kHz, transition band 20…24.1kHz (Δf=4.1kHz), stop band 24.1kHz…5.65MHz
- A digital FIR filter that meets these requirements would require more than $f_s/\Delta f$ = 11.3MHz/4.1kHz ≅ 2800 coefficients
  - Impractical!

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

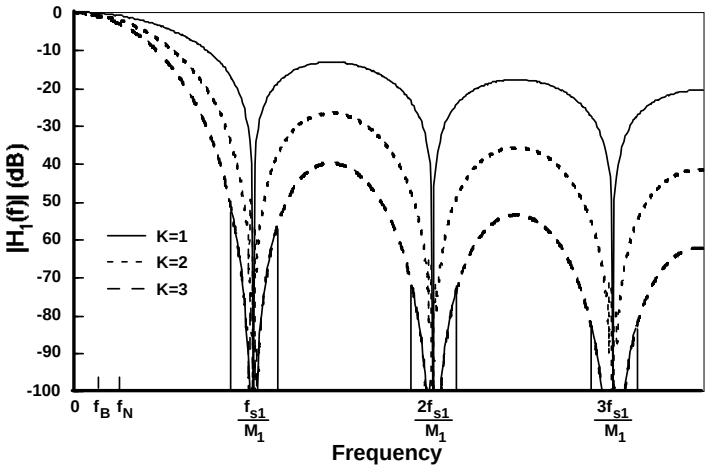
$$y(n)=\frac{1}{N}\sum_{i=0}^{N-1}x(n-i)$$

- 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!

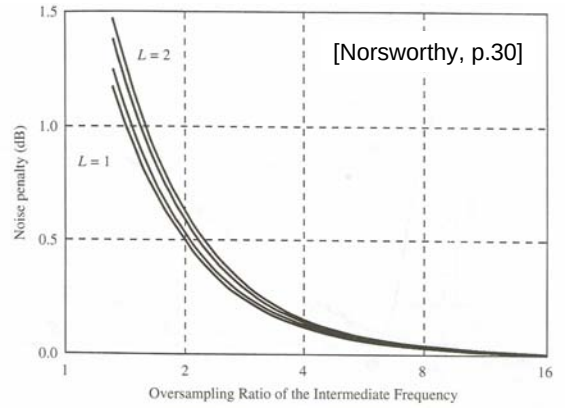
Cascade of K Sinc Filters



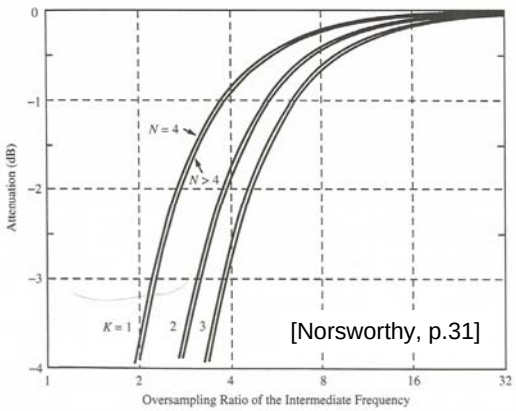
- Higher order means better rejection
  - But also more in-band droop
- Can show that for L^{th} order noise shaping, an $(L+1)^{th}$ order sinc filter is the best choice

Sinc Filter Performance

Noise penalty relative to "brick wall" filter



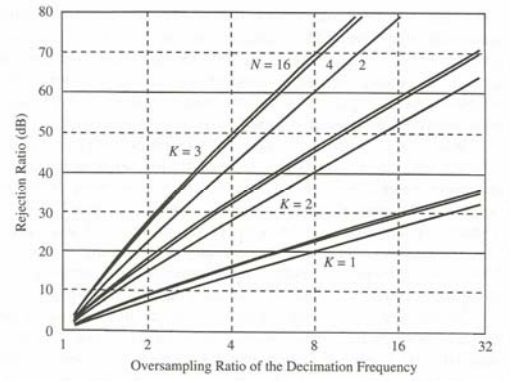
Droop



- 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

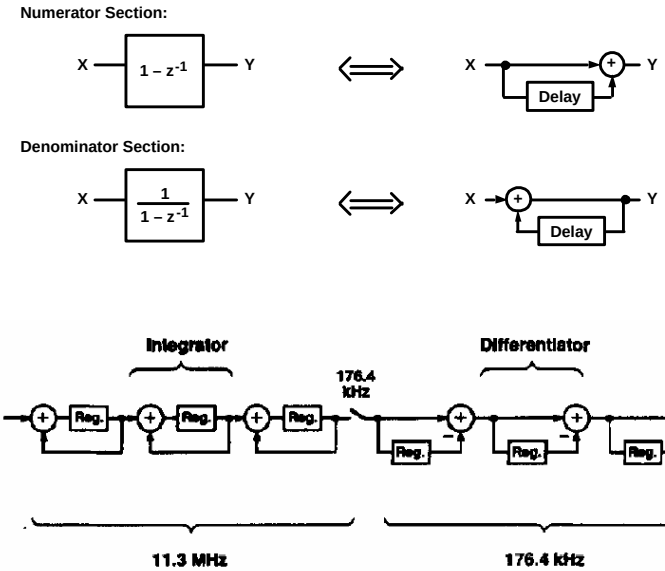
Sinc Filter Performance (2)

- 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

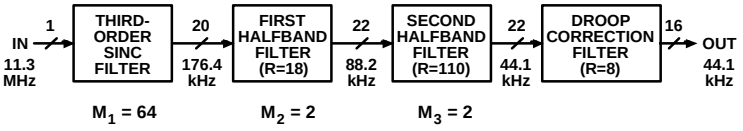


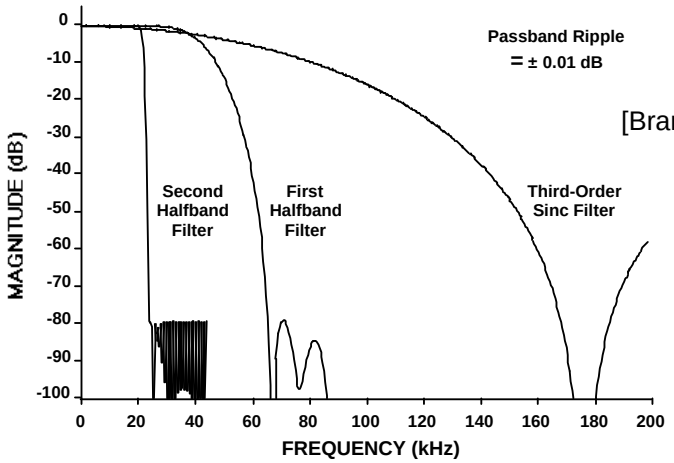
[Norsworthy, p.31]

Sinc Filter Implementation



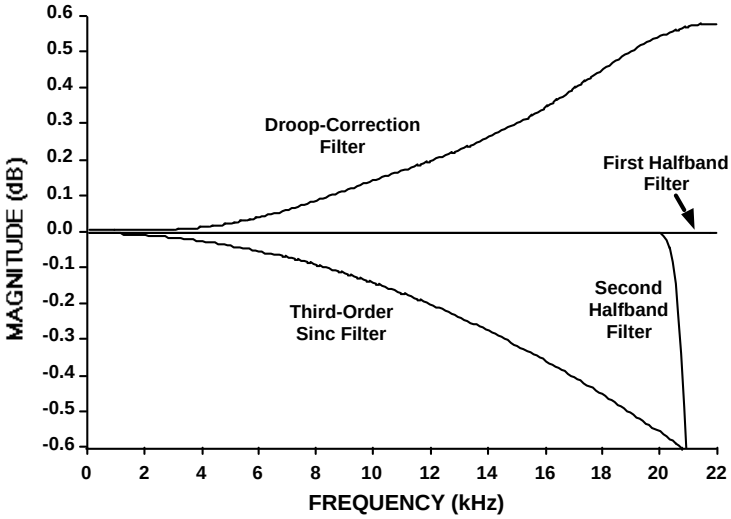
Complete Filter Implementation



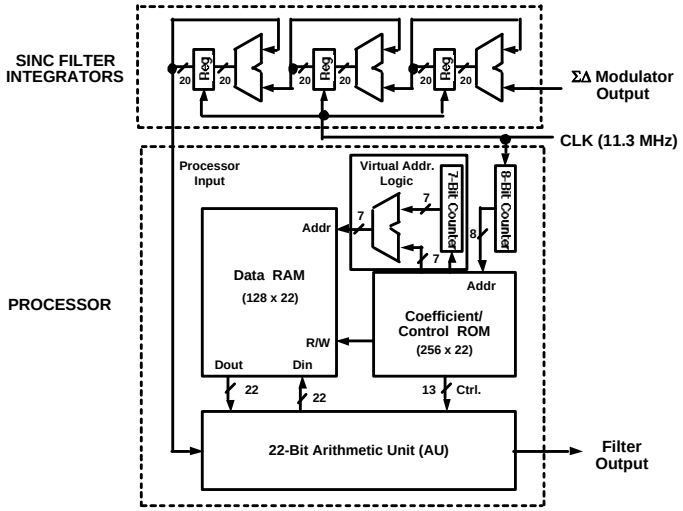


[Brandt & Wooley, JSSC 6/1994]

Droop Correction

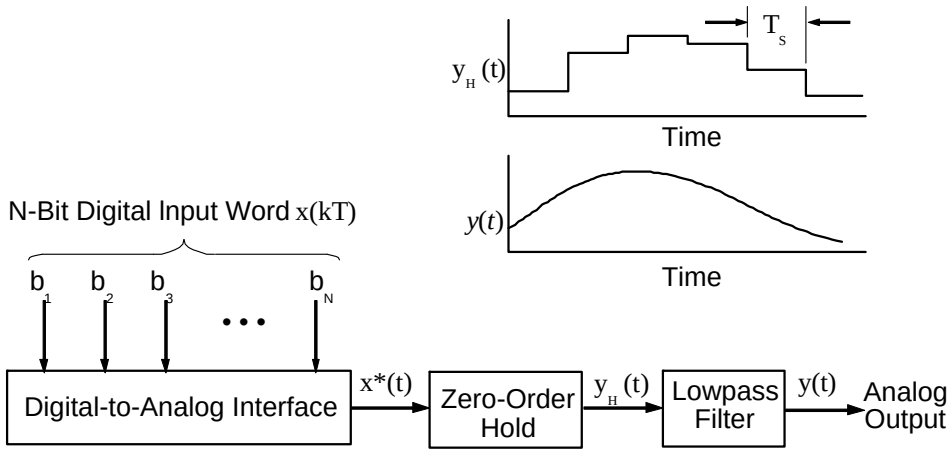


Implementation

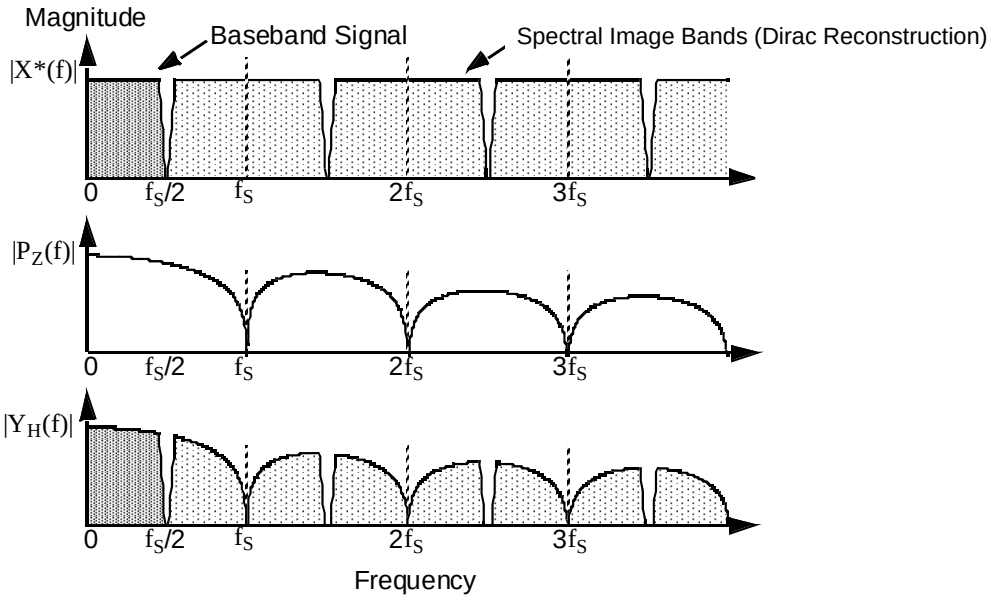


- 43 multiplications and 84 additions per output sample
- Can use serial arithmetic to minimize hardware area
  - Since output rate is usually fairly low

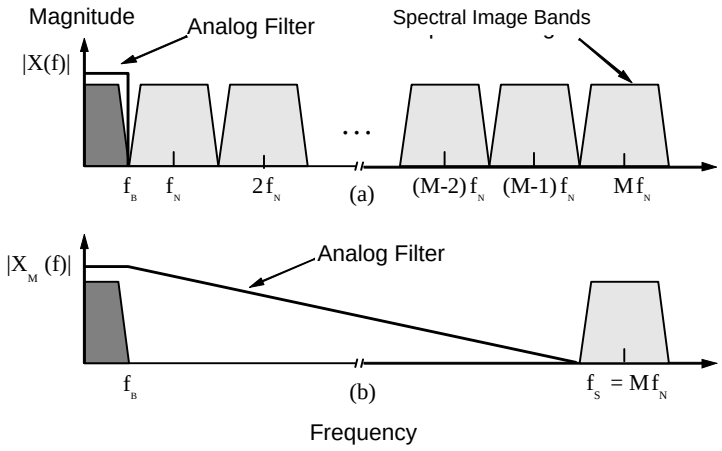
D/A Conversion Revisited



Frequency Spectra



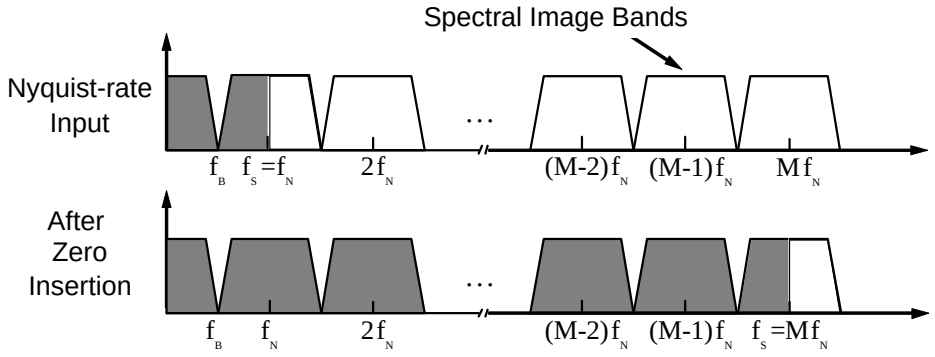
Oversampling



- Oversampling greatly reduces reconstruction filter requirements
- How to create oversampled DAC input from a Nyquist rate signal?

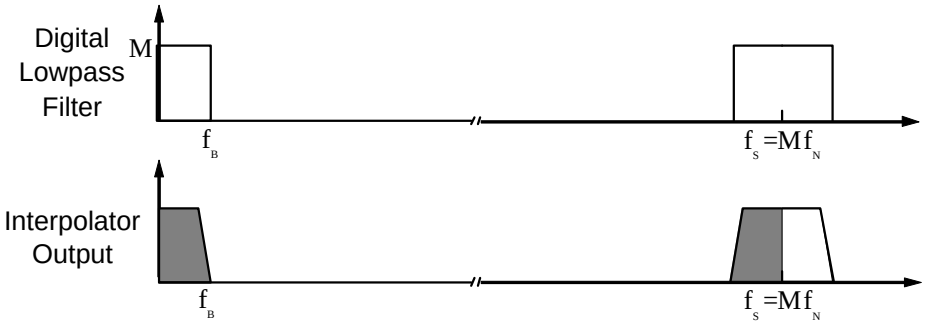
Interpolation (1)

- 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

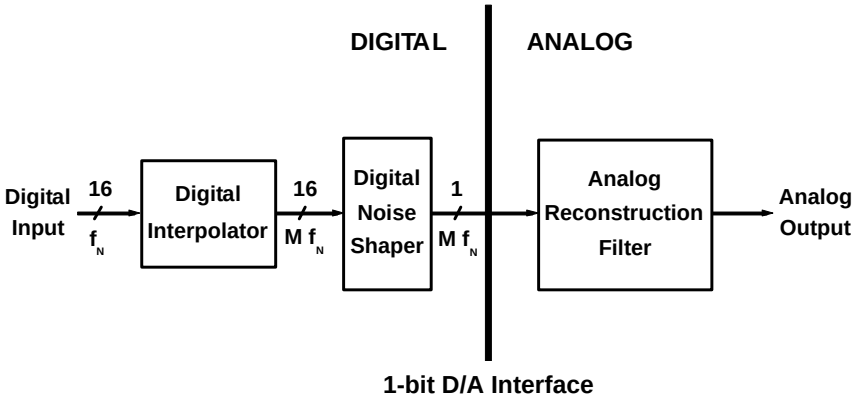


Interpolation (2)

- 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

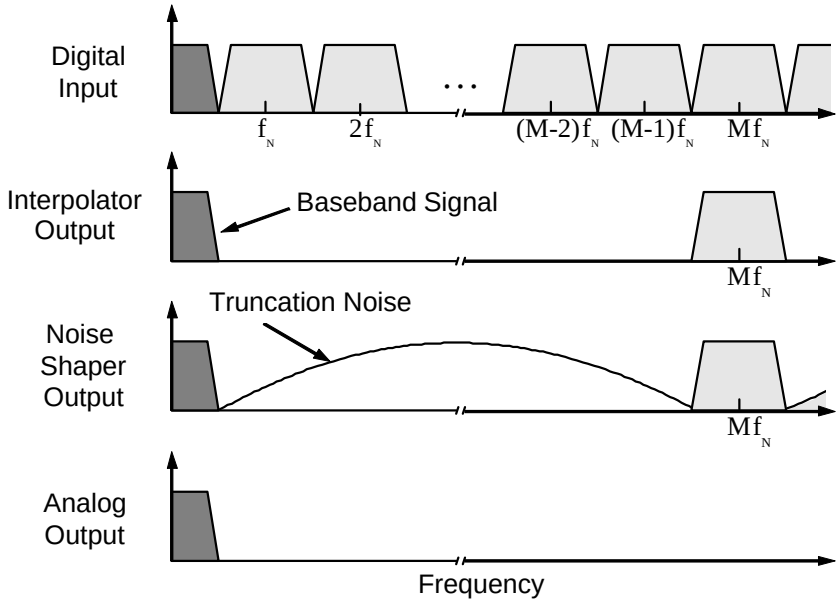


Example

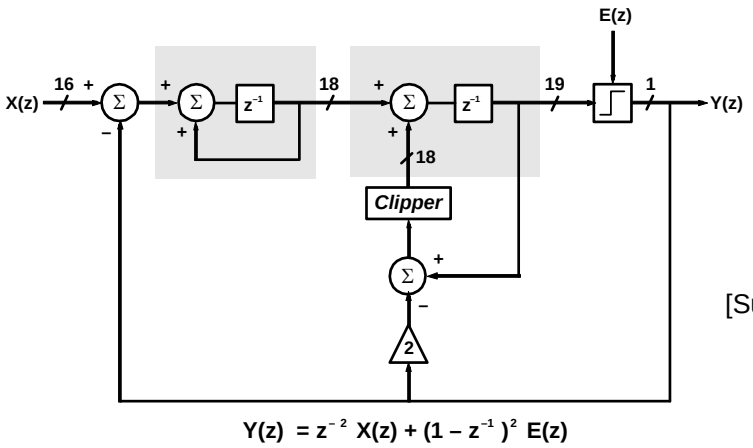


- Digital noise shaper is essentially a digital sigma-delta loop
  - Shapes "truncation noise" that results from truncating 16-bit word to a 1-bit output

Spectra



Example



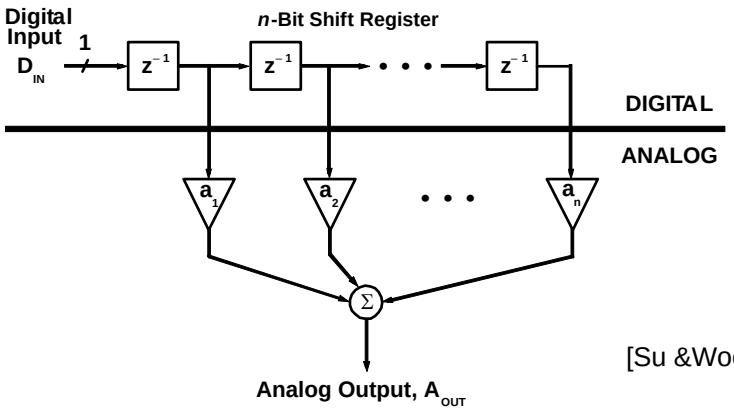
To 1-bit D/A Interface

[Su &Wooley, JSSC 12/94]

$$Y(z) = z^{-2} X(z) + (1 - z^{-1})^2 E(z)$$

- Clipper prevents second integrator from overflowing
  - Digital "wrap around" would cause large errors

Semi-Digital Reconstruction (1)

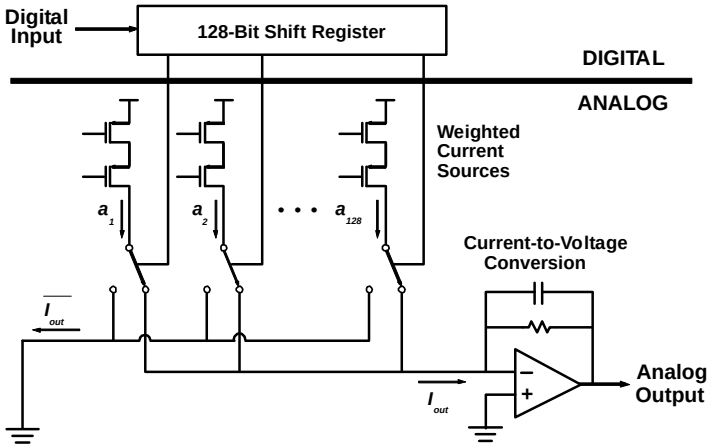


[Su &Wooley, JSSC 12/94]

$$H_{FIR}(z) = a_1 z^{-1} + a_2 z^{-2} + \ldots + a_n z^{-n}$$

- Attractive alternative to fully analog reconstruction filter
  - Build FIR filter with weighted analog outputs

Semi-Digital Reconstruction (2)

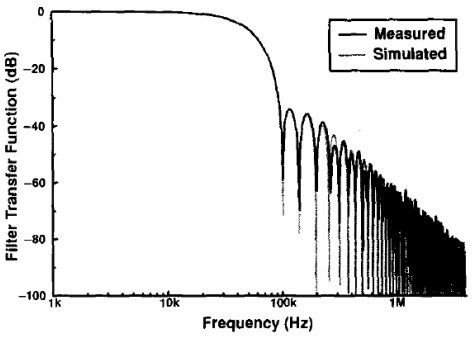


$$A_{OUT}(z) = \underbrace{\left[a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + \ldots + a_n z^{-n}\right]}_{H(z)} D_{IN}(z)$$

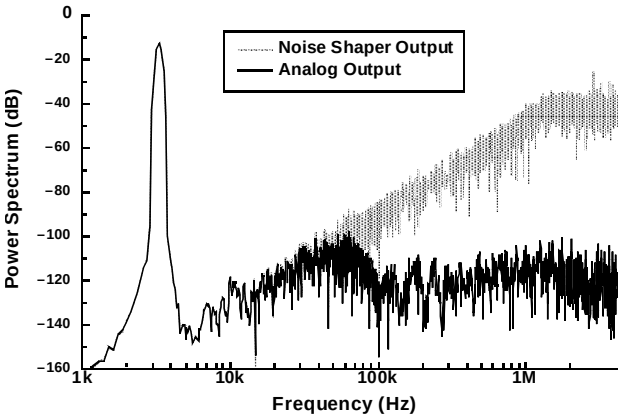
- Linear if H(z) is independent of $D_{IN}(z)$

Measurement Results

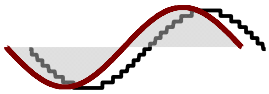
Reconstruction Filter



Output



Lecture 18
ADC Figures of Merit
Limits on ADC Power Dissipation



Boris Murmann
Stanford University
murmann@stanford.edu

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?

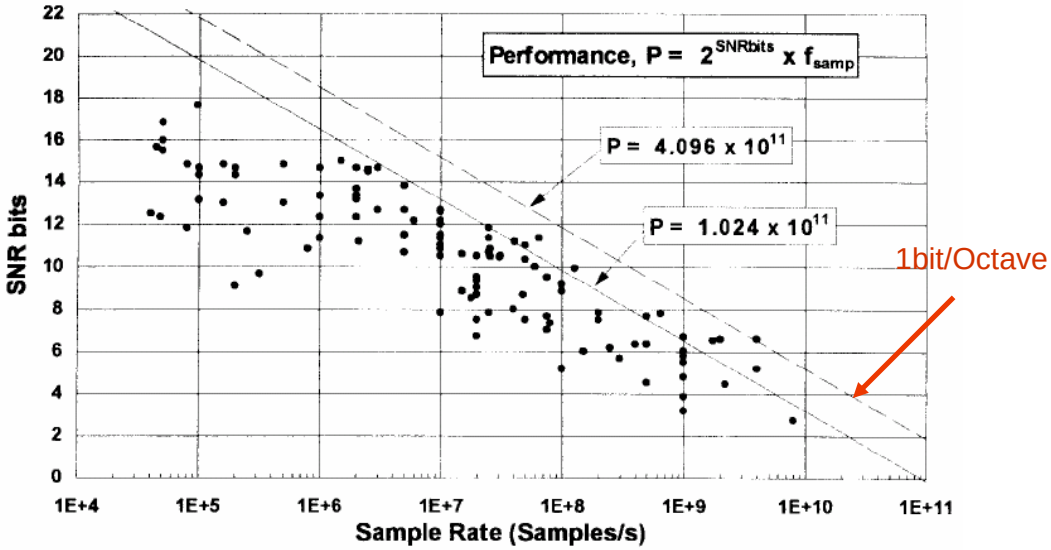
ADC Figures of Merit (2)

$$FOM_1 = f_s \cdot 2^{ENOB}$$

[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 ~ Gm ~ 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)

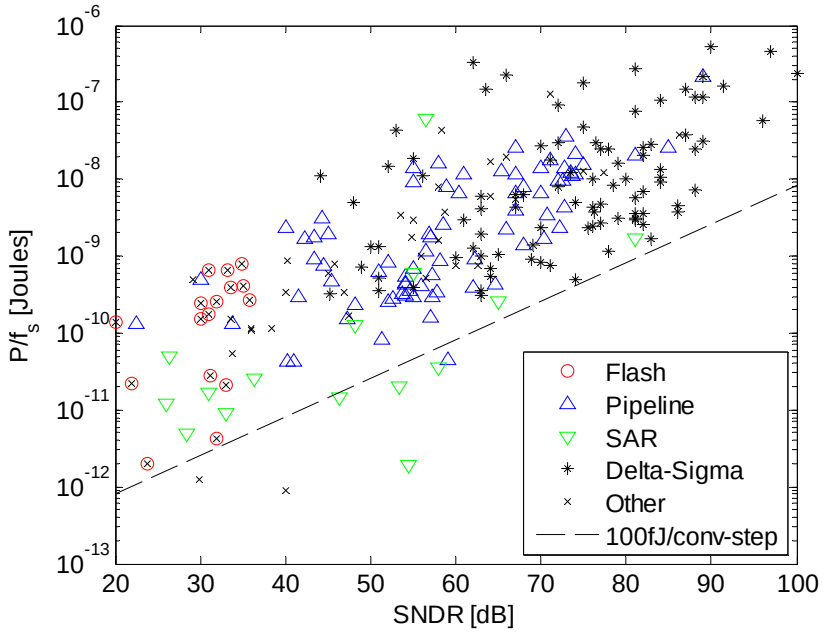
- FOM_2 in inappropriate for comparing ADCs that are limited by matching or thermal noise
  - Still the most widely used FOM in publications...
- "Tends to work" because not all power in an ADC is noise limited
  - E.g. Digital power, biasing circuits, etc.
- To better capture the case of noise limited 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)

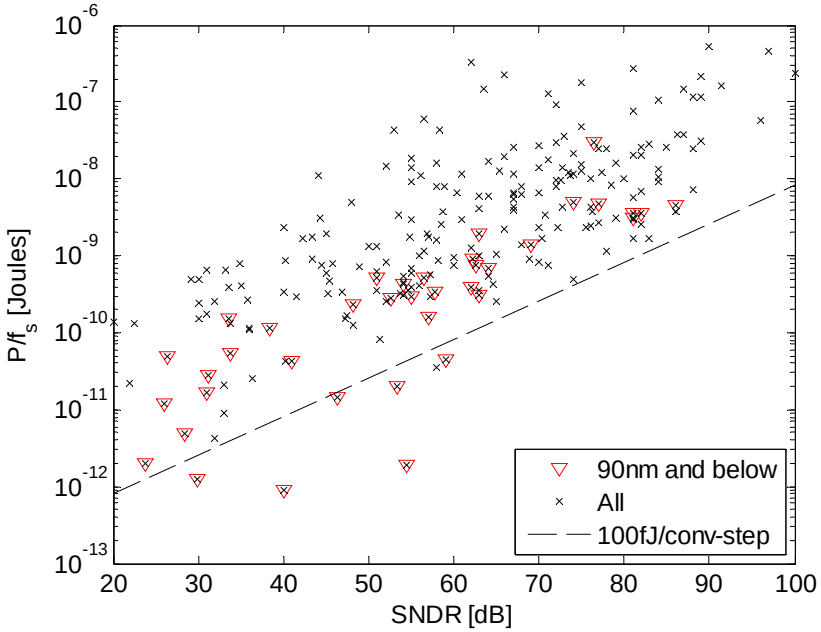
$$FOM_3 = \frac{Power}{2 \cdot Conversion\ Bandwidth} = "Energy\ per\ Nyquist\ Sample"$$

- 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

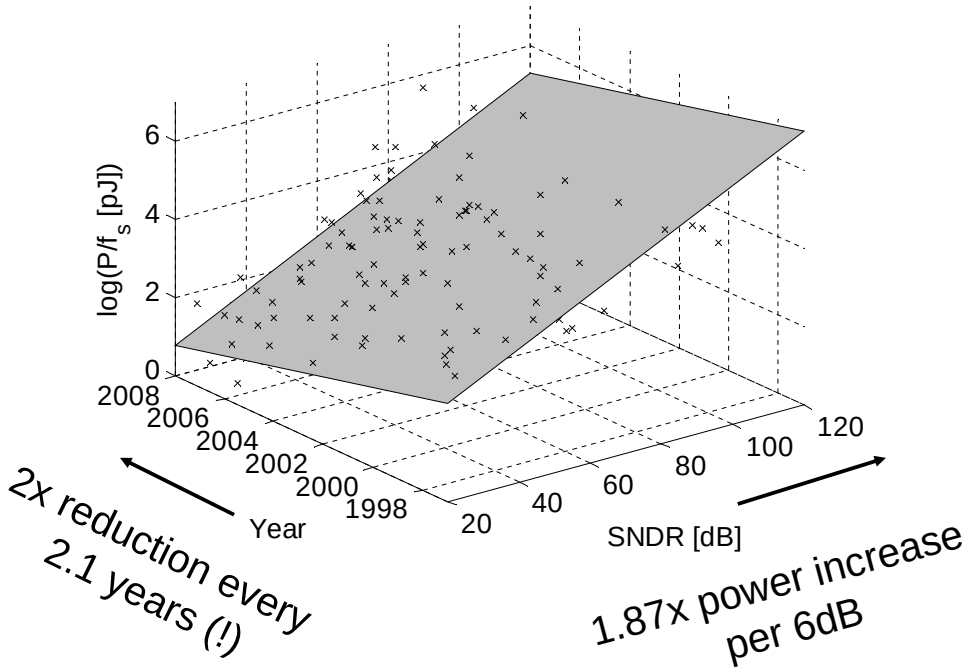
FOM3 (ISSCC & VLSI 1997-2008)



Power Dissipation in Sub-100nm CMOS



Power Dissipation Trend



Fundamental Limits

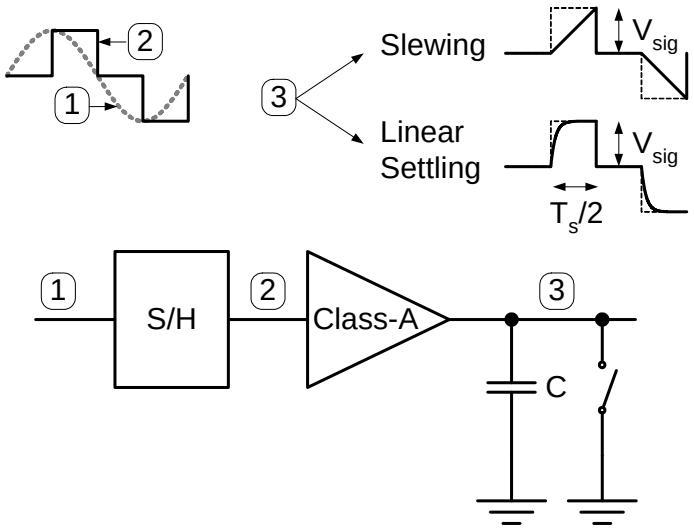
- Fundamental power limit for a class-B amplifier driving a single capacitor [Vittoz, ISCAS 1990]

$$P = 8 \cdot f_{sig} \cdot CV_{sig}^{\ 2} \qquad V_n^2 = \frac{k_B T}{C} \qquad SNR = \frac{0.5 \times V_{sig}^2}{V_n^2}$$

$$\boxed{\therefore P = 8 k_B T \cdot SNR \cdot f_{sig}}$$

- 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

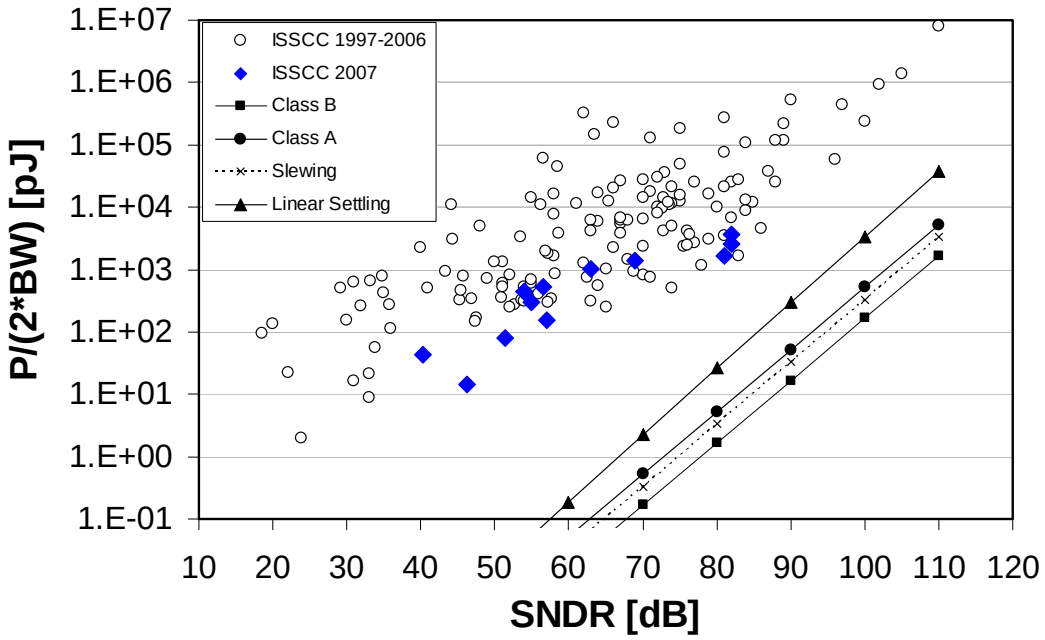
$$I_{bias} = C \cdot \left.\frac{dV}{dt}\right|_{max} = C \cdot \left.\frac{d}{dt}\right|_{max} \left[V_{sig}\left(1 - e^{-t/\tau}\right)\right] = C \cdot \frac{V_{sig}}{\tau}$$

Number of settling time constants: $N = \frac{T_s / 2}{\tau}$

$$\boxed{\therefore P = 16 \cdot N \cdot k_B T \cdot SNR \cdot f_{sig}}$$

- Much worse
  - E.g. N=6.9 for settling to 0.1% precision

Limit Lines



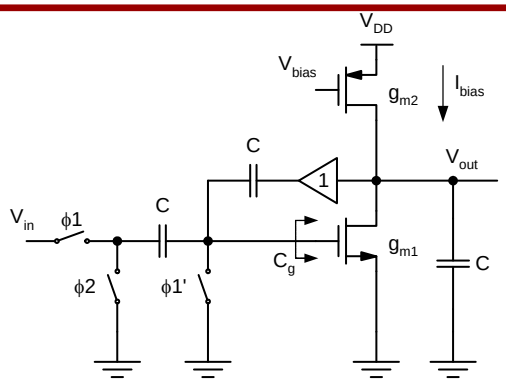
Discussion

- 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

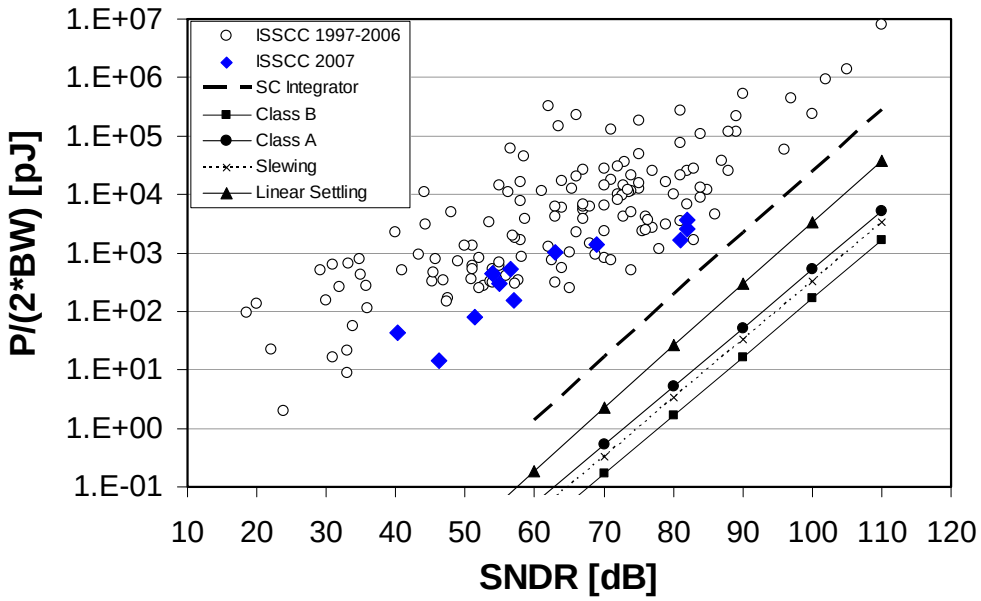
High SNR SC-Stage (2)

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)

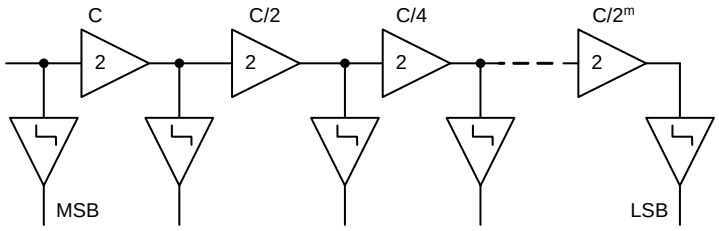


- Close to experimental data at high SNDR!

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



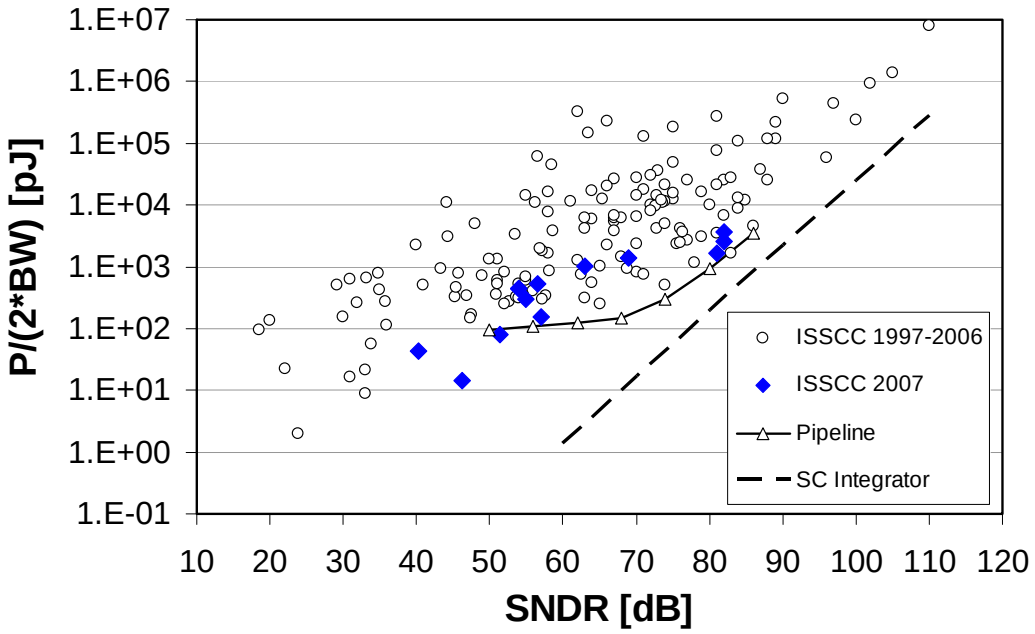
- Theoretical near optimum power scaling
  - Scale capacitance by gain of preceding stage
  - Stage 1 consumes half of total power
  - Adding one bit means power goes up 4x
- Caveat
  - Usually impractical to scale capacitors down to $C/2^m$

Stage Scaling Example

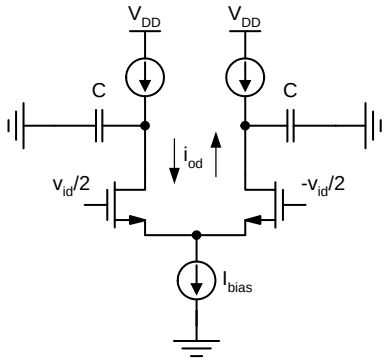
Number of Amplifiers	**13**	**12**	**11**	**10**
Stage Capacitances	1	1/4	1/16	1/64
	1/2	1/8	1/32	1/128
	1/4	1/16	1/64	*1/128*
	1/8	1/32	1/128	*1/128*
	1/16	1/64	*1/128*	*1/128*
	1/32	1/128	*1/128*	*1/128*
	1/64	*1/128*	*1/128*	*1/128*
	1/128	*1/128*	*1/128*	
	1/128	*1/128*		
	1/128			
ΣC	2.03	0.54	0.17	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



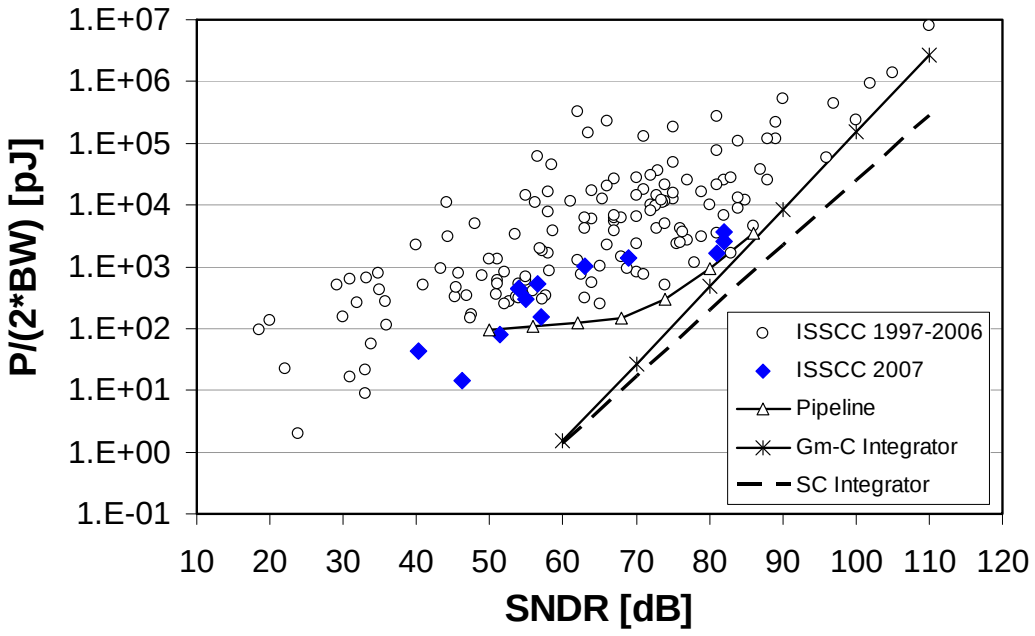
$$IM_3 \cong \frac{3}{32}\left(\frac{v_{id,max}}{V_{ov}}\right)^2$$

$$\eta_{cur} = \frac{i_{od,max}}{I_{bias}}$$

$$\eta_{cur} \cong \frac{v_{id,max}}{V_{ov}} = \sqrt{\frac{32}{3} IM_3}$$

- 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]

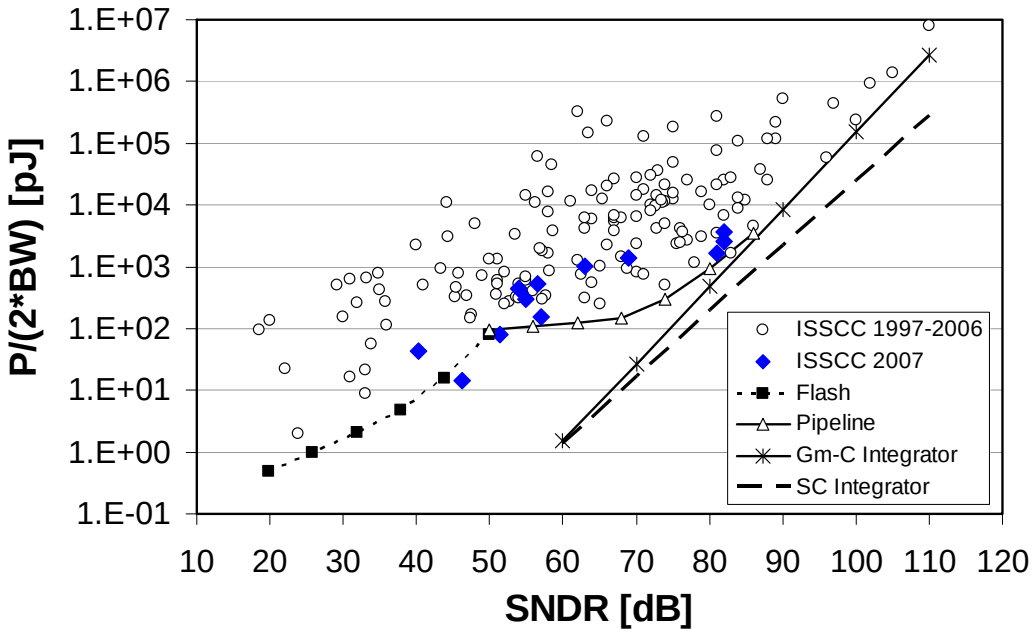
$$P = 24 \cdot C_{ox} \cdot A_{VT}^2 \cdot f_{sig} \cdot \left(\frac{V_{sig,rms}}{3 \cdot \sigma_{Vos}} \right)^2$$

- 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

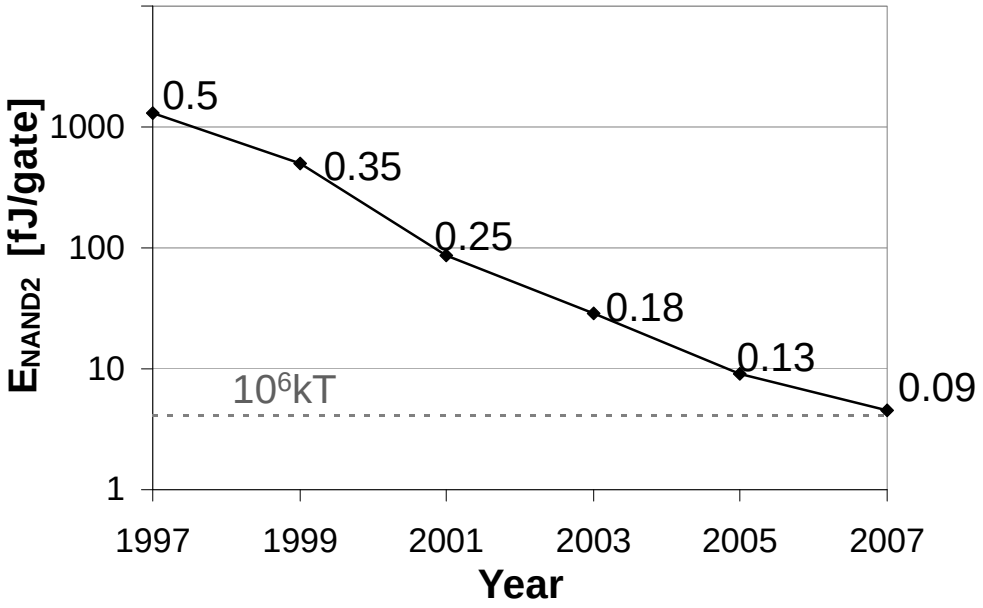
- Shown results include only minor assumptions about technology
- Scaling brings some good, some bad news offsetting each other
  - – Lower V_{DD}, lower V_{swing}/V_{DD}, …
  - + Lower E_{dyn}, higher f_t enables moderate/weak inversion operation with high g_m/I_D,…
- Limit lines won't move much, unless someone hands us a new disruptive technology

Future Opportunities

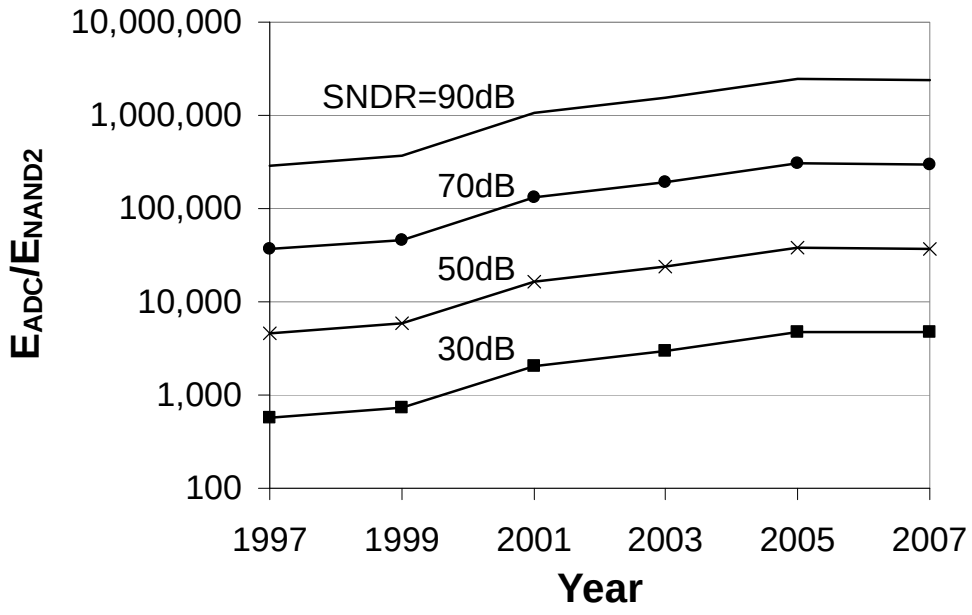
- More intelligent ADCs
  - Improved average power dissipation by adapting to instantaneous speed/resolution requirements
- "Minimalistic" ADCs using significantly simpler circuits
  - Digital compensation of resulting non-idealities
  - Digital postprocessing is (within limits) "free" in terms of area 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)

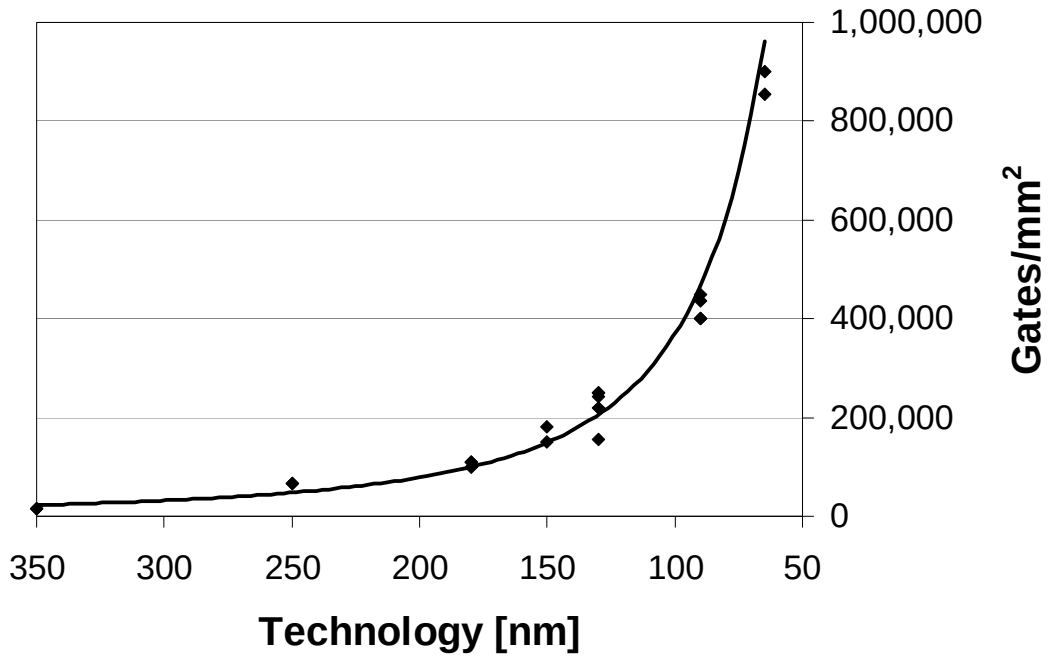
SNDR	E_{ADC}/E_{NAND2}
30	4,679
50	37,432
70	299,479
90	2,396,045

⇨ Additional digital processing is costly!

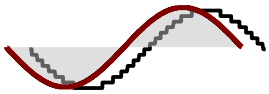
⇨ Several tens of thousand gates are "free"

⇨ Use as many gates as you can fit…

Digital Logic Gate Density Trend

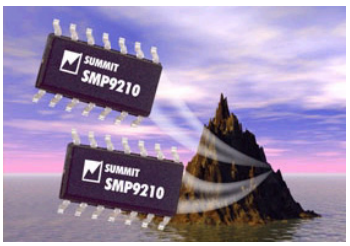


Data Converter Testing



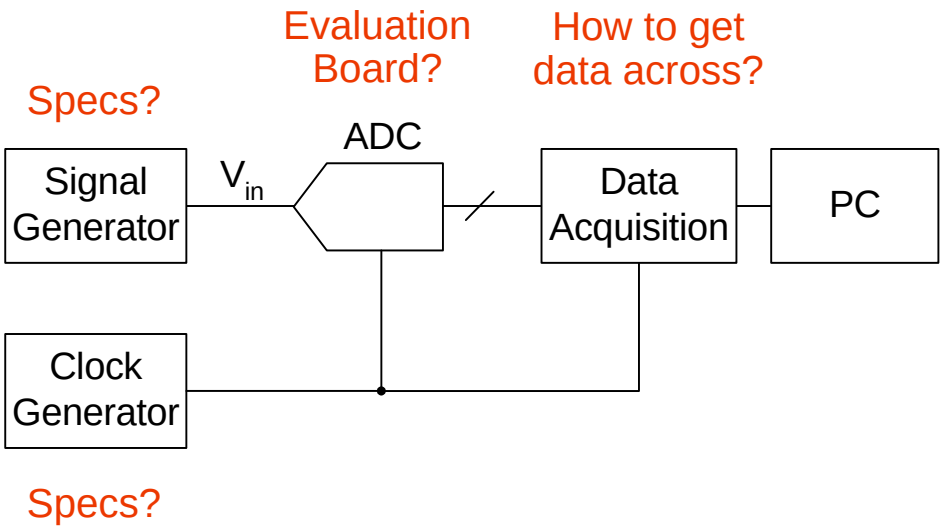
Boris Murmann
Stanford University
murmann@stanford.edu

Just Got Silicon Back...



- Now what ?
- Practical aspects of converter testing
- Equipment requirements
- Pitfalls

ADC Test Setup



State-Of-The-Art ADC (2001)

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 @ $f_{in}=f_s/2=37.5MHz$
- Let's see, how about the "value priced" signal generator we have in the lab...

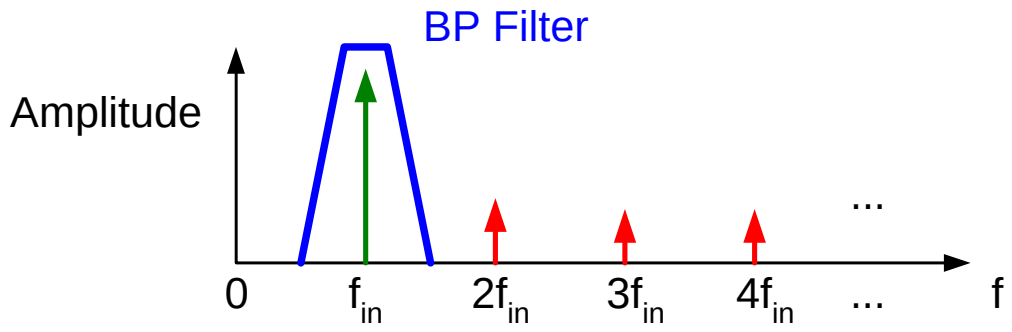
- f=0...15MHz
- Harmonic distortion (f>1MHz): -35dBc
- Need something better...

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

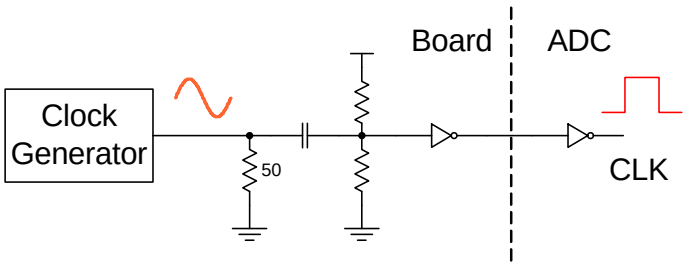
K&L Model	Frequency Range (MHz)	Passband Insertion Loss	Length Inch/mm	Width Inch/mm	Height Inch/mm
5BT-30/76-5-N/N	30-76	1.3 dB Max	9.80/249	5.38/137	2.75/50
5BT-63/125-5-N/N	63-125	1.3 dB Max	9.80/249	5.38/137	2.75/50
5BT-125/250-5-N/N	125-250	1.3 dB Max	9.80/249	5.38/137	2.75/50
5BT-250/500-5-N/N	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.

$$x(t) = A\cos(\omega_0 t + \theta) \quad (3)$$

$$y_k = A\cos(\omega_0 k T_s + \theta) + g(x(t))\,|_{t=kT_s} - A\omega_0 \Delta_k \sin(\omega_0 k T_s + \theta) + n_k \quad (4)$$

After removal of harmonics:

$$e_k = -A\omega_0 \Delta_k \sin(\omega_0 k T_s + \theta) + n_k \quad (5)$$

Spectrum of squared sequence contains a tone proportional to jitter:

$$\begin{aligned} E\{e_k^2\} &= E\{A^2\omega_0^2\Delta_k^2 \sin^2(\omega_0 k T_s + \theta) + n_k^2\} \\ &= E\left\{\left(\frac{A^2\omega_0^2\Delta_k^2}{2} + n_k^2\right) - \frac{A^2\omega_0^2\Delta_k^2}{2}\cos(2\omega_0 k T_s + 2\theta)\right\} \quad (6) \\ &= \left(\frac{A^2\omega_0^2\sigma_\Delta^2}{2} + \sigma_n^2\right) - \frac{A^2\omega_0^2\sigma_\Delta^2}{2}\cos(2\omega_0 k T_s + 2\theta) \quad (7) \end{aligned}$$

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

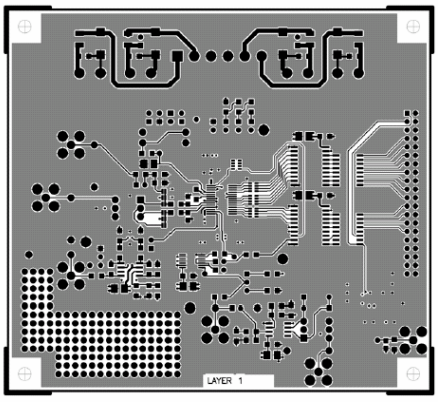


Figure 21. TSSOP Evaluation Board Layout, Primary Side

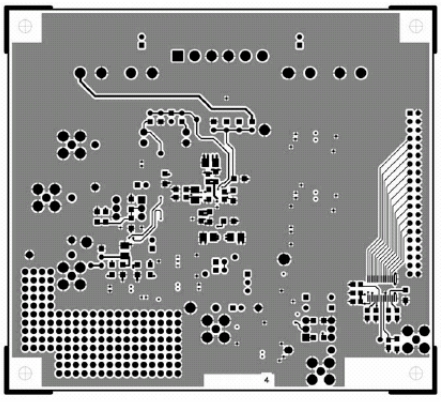


Figure 22. TSSOP Evaluation Board Layout, Secondary Side

[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!

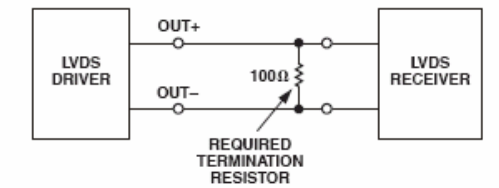


Figure 1. LVDS Output Levels

Analog Devices Application Note 586**:** "LVDS Data Outputs for High Speed ADCs"

LVDS Outputs

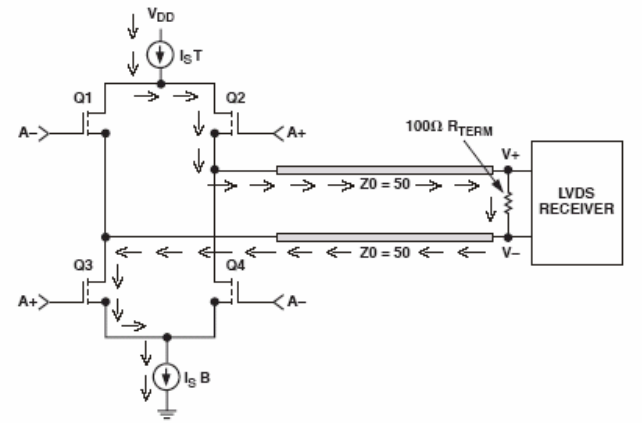
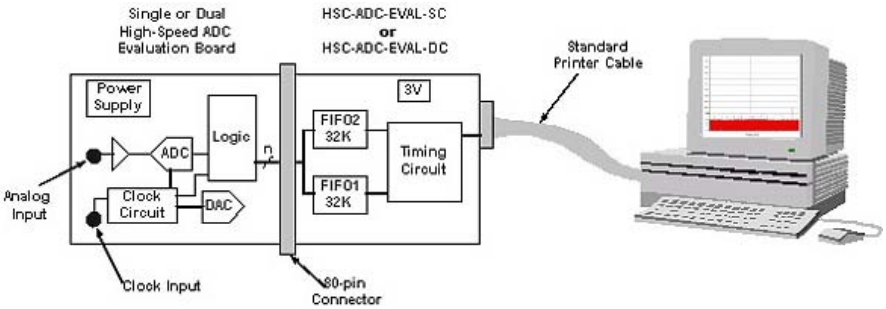


Figure 4. LVDS Output Current

Analog Devices Application Note 586**:** "LVDS Data Outputs for High Speed ADCs"

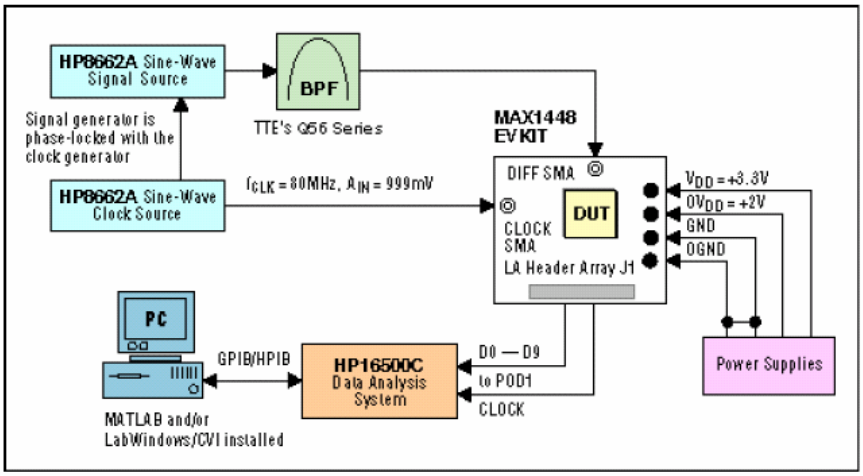
Data Acquisition

- 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

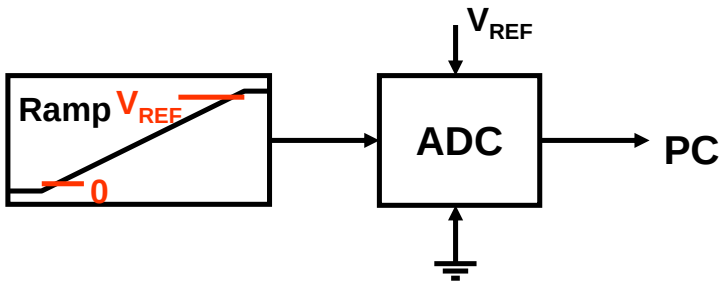


[Maxim Application Note 729**:** "Dynamic Testing of High-Speed ADCs, Part 2]

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

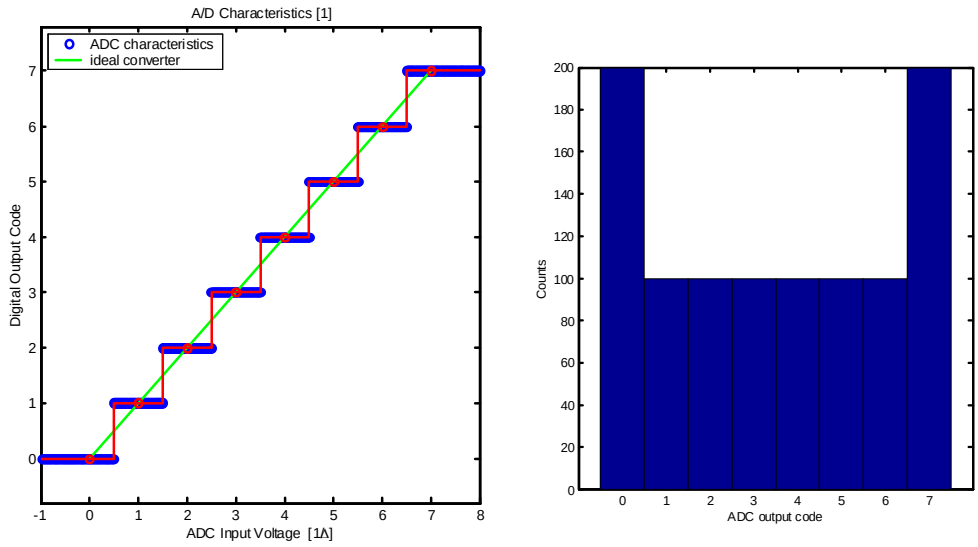
Basic Histogram Test Setup



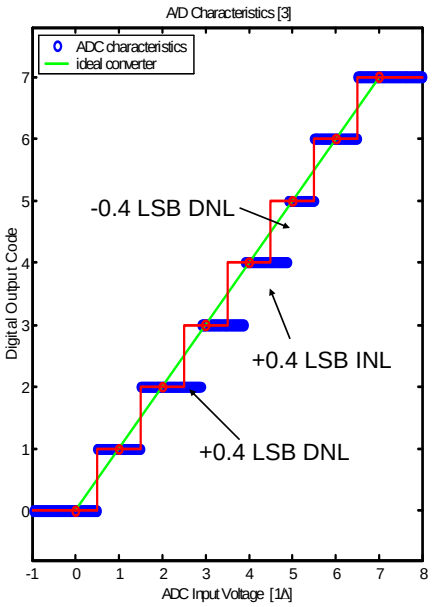
- DNL follows from total number of occurrences of each code
- Ramp speed is adjusted to provide e.g. an average of 100 outputs of each ADC code (for 1/100 LSB resolution)
- Ramps can be quite slow for high resolution ADCs

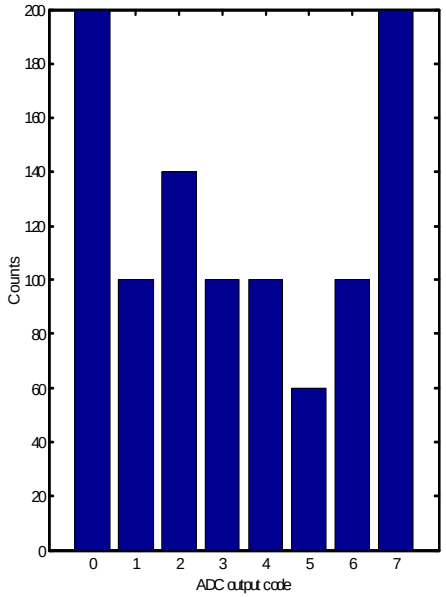
$$\frac{\text{(65,536 codes)(100 conversions/code)}}{\text{100,000 conversions/sec}} = 65.6 \text{ sec}$$

Histogram of Ideal 3 Bit ADC



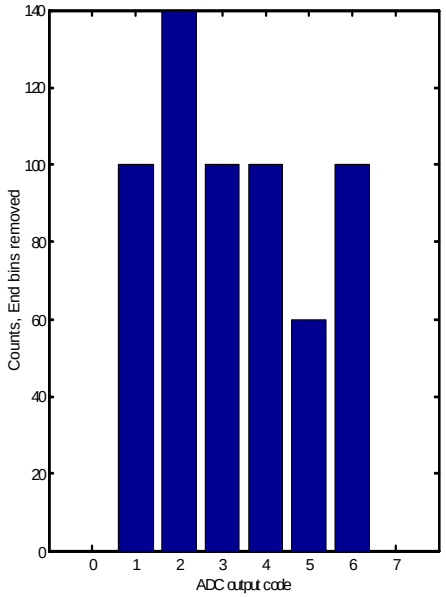
Histogram of Sample 3 Bit ADC





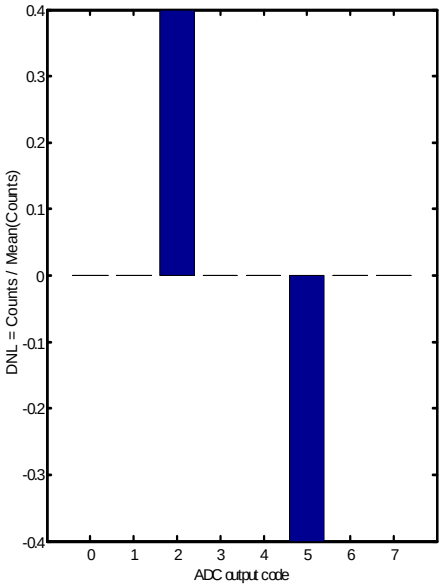
DNL from Histogram (1)

- Step 1
  - Remove “over-range bins” (0 and 7)



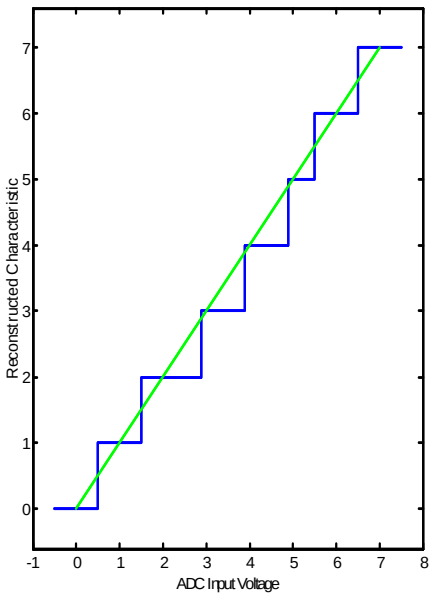
DNL from Histogram (2)

- 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

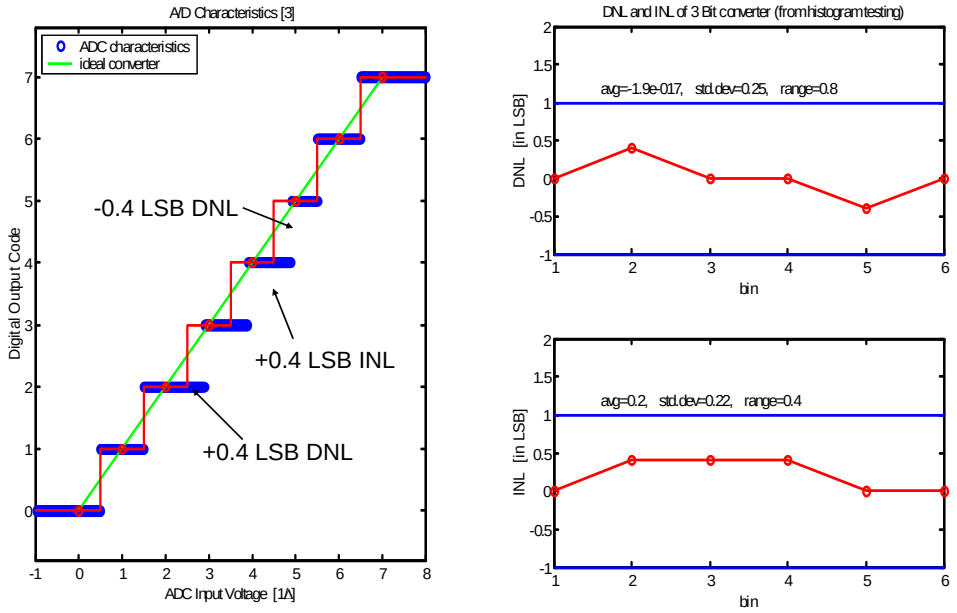


INL from Histogram

- 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

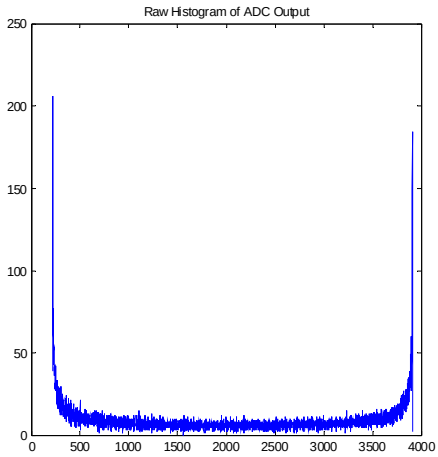


DNL and INL of Sample ADC

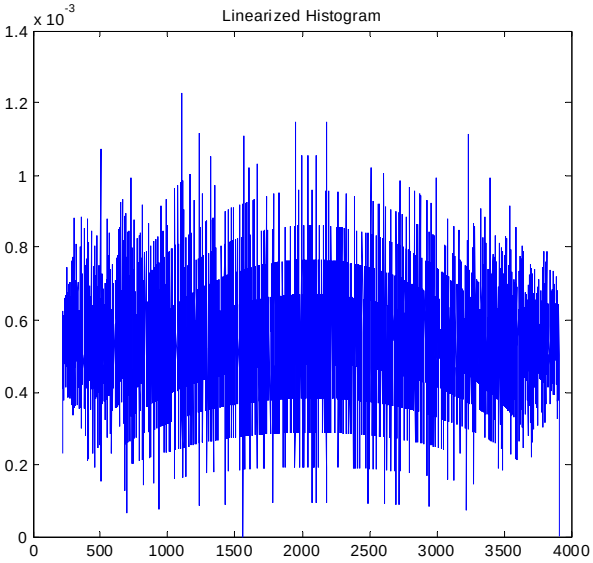


Sinusoidal Inputs

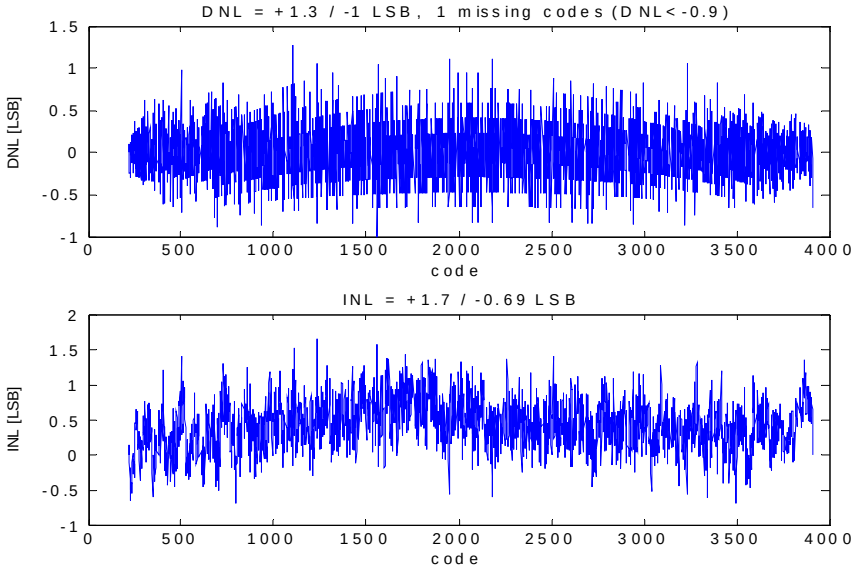
- Precise ramps are hard to generate
- Solution
  - Use sinusoidal test signal
- Problem
  - Ideal histogram is not flat but has “bath-tub shape”



After Correction for Sinusoidal pdf



Resulting DNL and INL



Correction for Sinusoidal pdf

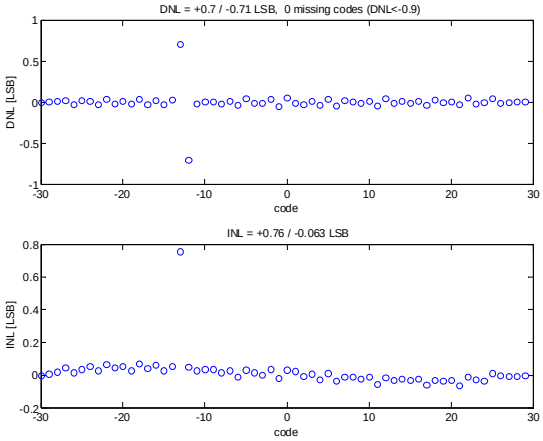
- References
  - M. V. Bossche, J. Schoukens, and J. Renneboog, “Dynamic Testing and Diagnostics of A/D Converters,” IEEE TCAS, Aug. 1986.
  - IEEE Standard 1057
- Is it necessary to know the exact amplitude and offset of the sine wave input?
  - No!
- There exists a great deal of confusion about this 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);
```

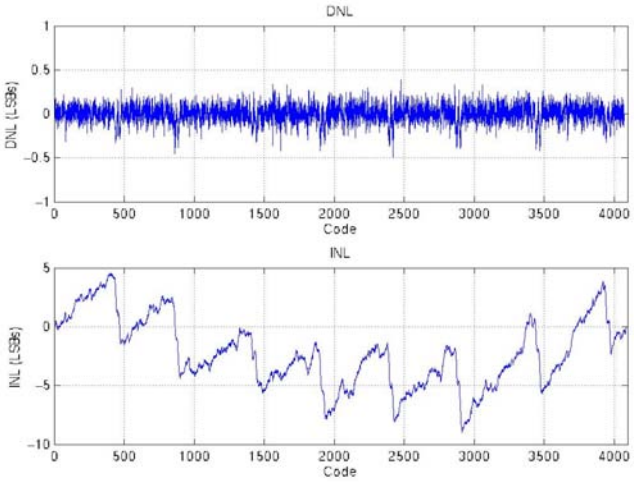


Limitations of Histogram Testing

- 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.

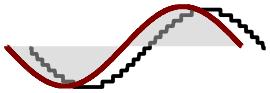
Hiding Problems in the Noise

- 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]

Layout Considerations



Boris Murmann
Stanford University
murmann@stanford.edu

Overview

- Impact of device mismatch
  - Linearity of mulit-bit DACs, uncalibrated pipeline ADCs, …
  - Finite common mode and supply rejection
  - Offset and offset drift; important e.g. in bandgap references
- Noise and decoupling
  - Capacitive coupling, inductive coupling (bond wires)
  - Supply coupling
    - Separate supplies for analog and digital
  - Substrate coupling
- Floorplanning
  - Organize the layout to minimize device mismatch and coupling effects

Device Mismatch Mechanisms

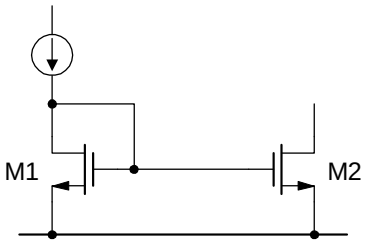
- Wafer-to-Wafer, Batch-to-Batch variations
- Spatial effects
  - Long distance
    - Gradients
  - Short distance
    - Statistics
- Circuit dependence
  - Differential structures
    - Differential pair
    - Current mirror
  - Bias
- Layout dependence

References

- M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," IEEE J. of Solid-State Circuits, vol. 24, pp. 1433-1439, October 1989.
  - Mismatch model
  - Statistical data for 2.5μm CMOS
- Jeroen A. Croon, Maarten Rosmeulen, Stefaan Decoutere, Willy Sansen, Herman E. Maes, "An easy-to-use mismatch model for the MOS transistor," IEEE J. of Solid-State Circuits, vol. 37, pp. 1056 - 1064, August 2002.
  - 0.18μm CMOS data
  - Qualitative analysis of short-channel effects on matching
- C. H. Diaz *et al.*, "CMOS technology for MS/RF SoC," *IEEE Trans. Electron Devices*, pp. 557-566, March 2003.
  - More recent matching data

Mismatch Modeling

Experiment:



$$\frac{\Delta I_D}{I_D} = 1\%$$

- Experimental result applies to one particular configuration
- What about:
  - Device size
    - W
    - L
    - Area
  - Bias
    - V_{GS}
  - Physical proximity
  - …
- Need parameterized model

Mismatch Model

$$\sigma^2(\Delta P) = \frac{A_P^2}{WL} + S_P^2 D_x^2$$

$\sigma^2(\Delta P)$: *variance of paramter P*

WL: *device area*

D_x: *distance between device centers*

A_P: *measured area parameter*

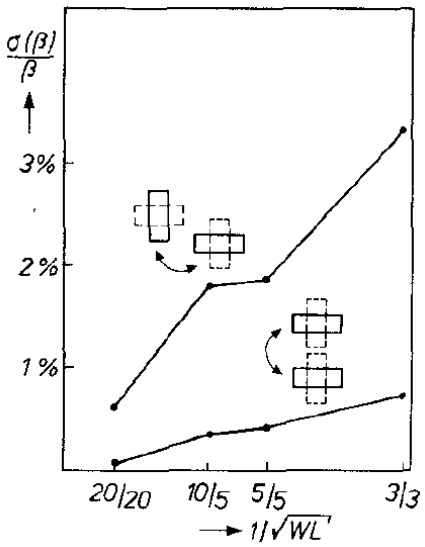
S_P: *measured distance parameter*

- Second term due to distance parameter is usually small
  - Unless D_x >> 10…100μm
- Can use "common centroid layout" to make D_x=0
  - Helps cancel process gradients
- Assuming that we've done everything right, we are left with local random variations, governed by A_P

Basic Rules for Matching

- Use the same W and L and use M unit devices to generate current ratios (takes out ΔW and ΔL effects)
- Use M factors that are even, preferably factors of 4 (to avoid anisotropy effects)
- Use common-centroid, or nearly common-centroid, layout (takes out systematic gradients, e.g. oxide thickness and doping)
- Use dummy devices at the edges of the array (takes out etch loading effects)
- Keep matched devices away from power sources (>50mW)
- Ensure clean and well balanced routing
  - Avoid having contacts/vias or irregular metal routing patterns over matching sensitive devices
- Route currents to bridge long distances, not voltages - IR drops can cause big systematic mismatches

Orientation Effects



- Si and transistors are not (perfectly) isotropic
  - Stress induced mobility variations: several percent error
  - Tilted wafers: ~5% error
- Make sure to have same direction of current flow in each device!

Common Centroid Layout

- Reference
  - Hastings, *The Art of Analog Layout*, Prentice Hall, 2001
- Determine groups of matched components
  - Depends on circuit function
    - All transistors in a mirror
    - Diff-pair and load in an amplifier
      - Should they be matched individually or jointly?
- Divide into segments
  - Based on unit elements, if there is a common divisor
  - Avoid small (<70%) fractional elements if no common divisor exists
    - Example: Need matching resistors of 39.7k and 144.5k
      - 144.5=3.68*39.7 (3 unit devices, plus 0.68*unit device)
      - 144.5=10.92*(39.7/3) (10 unit devices, plus 0.92*unit device; better choice)

Common Centroid Rules (1)

- Coincidence
  - Center of all matched devices should coincide, at least approximately
- Symmetry
  - Along X and Y axis
  - Symmetry lines of ABAB pattern do not line up!
- Dispersion
  - Segments of each device should be distributed throughout the array as uniformly as possible
    - Reduces sensitivity to higher order (nonlinear) gradients
  - One dimensional examples
    - ABBAABBA: 3 repetitions
    - ABABBABA: 1 repetition
      - Has higher dispersion (preferable)

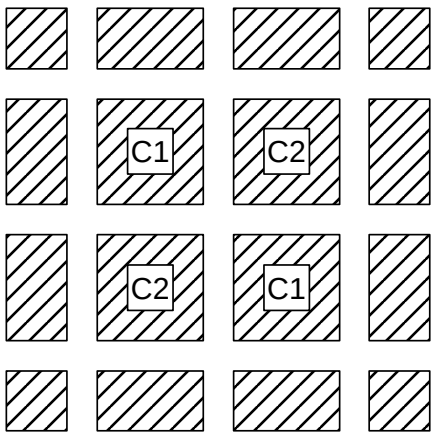
Common Centroid Rules (2)

- Compactness
  - Make array as compact as possible and approximately square
  - 2D patterns achieve best symmetry
    - X symmetry comes from interdigitation, and does rely on unit device symmetry
    - Example patters

$$\begin{matrix} {}_{D}A_{S}B_{D} \\ {}_{D}B_{S}A_{D} \end{matrix} \qquad \begin{matrix} {}_{D}A_{S}B_{D}B_{S}A_{D} \\ {}_{D}B_{S}A_{D}A_{S}B_{D} \end{matrix} \qquad \begin{matrix} {}_{D}A_{S}B_{D}B_{S}A_{D} \\ {}_{D}B_{S}A_{D}A_{S}B_{D} \\ {}_{D}A_{S}B_{D}B_{S}A_{D} \\ {}_{D}B_{S}A_{D}A_{S}B_{D} \end{matrix}$$

- In some cases, 2-D common centroid creates too much routing overhead, which violates rule of compactness
  - E.g. resistors, which are hard to arrange as "square" elements
  - Sometimes better off with simple 1-D pattern

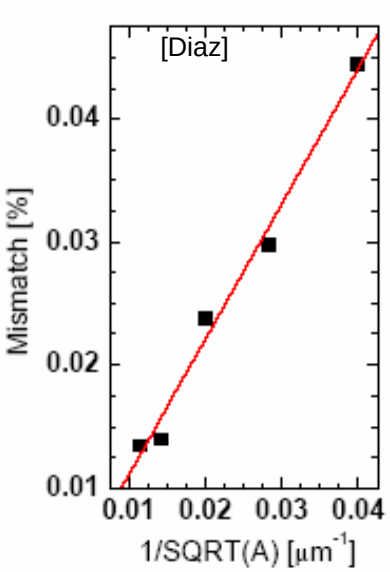
Dummy Cells



- Watch out for capacitor mismatch due to routing imbalance!

MIM Capacitor Mismatch

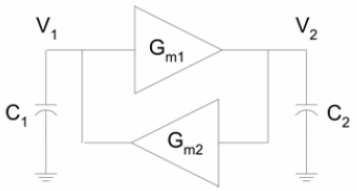
0.13μm CMOS process



- E.g. capacitor with A=33μmx33μm
  - C≅1.1pF
  - 1/sqrt(A)=0.03μm^{-1}
  - 3-σ Mismatch=0.03%

Routing Imbalance at Latch Output

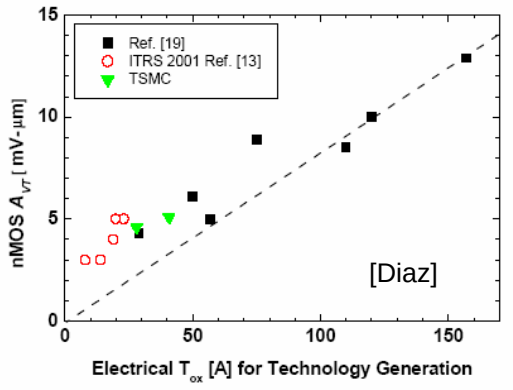
- Regenerative latch



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

V_t Mismatch

$$\sigma^2(\Delta V_{t0}) \cong \frac{A_{Vt}^2}{WL}$$



- In 0.18µm, T_{ox}=6.5nm, $A_{Vt} \cong$ 3mVµm
  - Means that a differential pair will have $\sigma(\Delta V_t)$ of about 3mV if the gate area of each transistor is 1µm^2
  - Again, this assumes that we've done a very good job in eliminating gradients and all other potential systematic errors

Example: Current Mirror

$$\frac{\Delta I_D}{I_D} = \frac{g_m}{I_D}\Delta V_{TH} + \frac{\Delta\beta}{\beta}$$

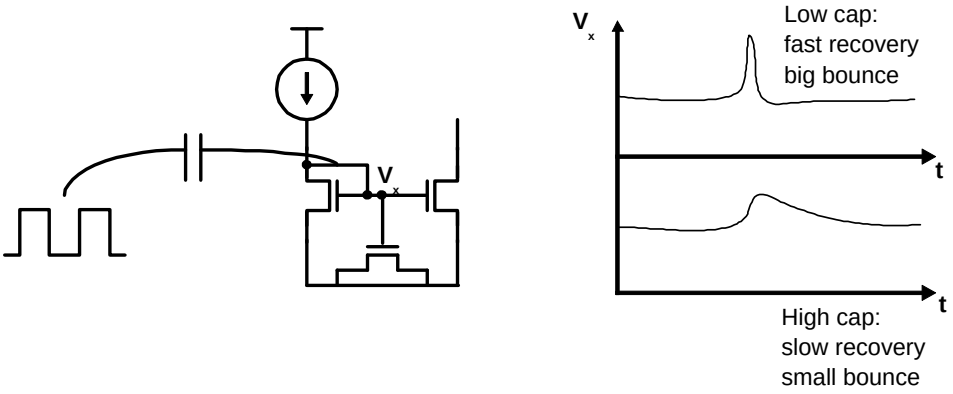
$$\sigma^2_{\Delta I_D/I_D} \cong \left(\frac{g_m}{I_D}\right)^2 \frac{A_{Vt}^2}{WL} + \frac{A_\beta^2}{WL}$$

- Example for 0.18µm technology: $A_{Vt} \cong$ 3mVµm, $A_\beta \cong$ 1%µm, W=10µm, L=0.18µm, g_m/I_D=10V^{-1}

$$\sigma_{\Delta I_D/I_D} = \sqrt{\frac{10^2}{V}\frac{(3mV)^2}{10\cdot 0.18} + \frac{(1\%)^2}{10\cdot 0.18}} = \sqrt{(2.2\%)^2 + (0.74\%)^2} = 2.32\%$$

- Lower g_m/I_D (higher V_{GS}-V_t) results in improved matching

Capacitive Coupling

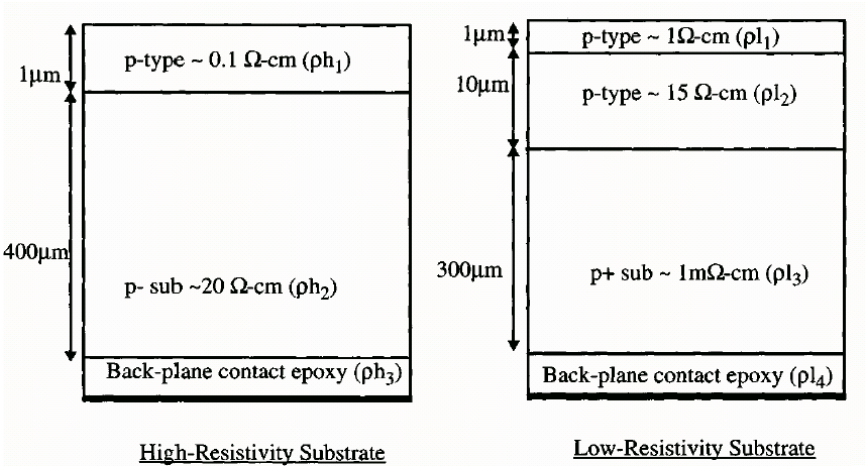


- Can use decoupling capacitors to reduce the amplitude of noise coupling into bias nodes
- If noise is "deterministic" and occurs at the right point in time, you might be better off not decoupling, but making the bias node "fast" so it can recover quickly!

Shielding

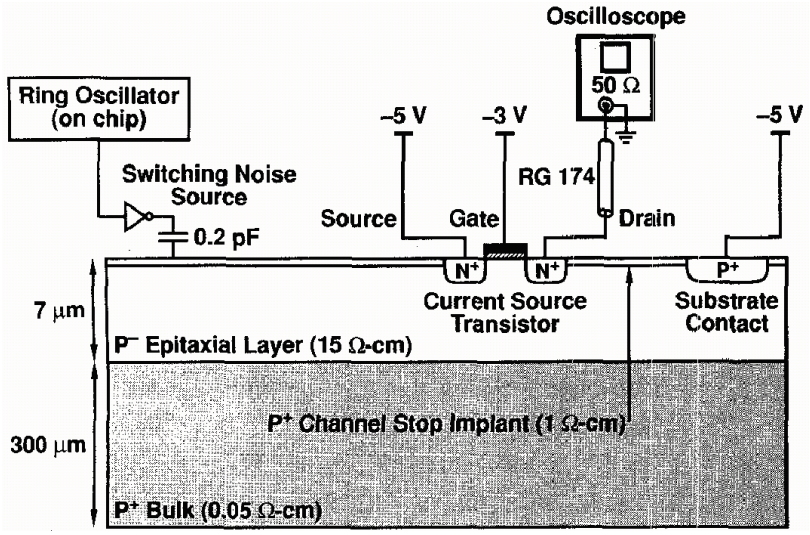
- Can attenuate capacitive coupling by shielding sensitive signals with traces running along their side, or underneath
  - Usually creates additional capacitive load!
- For differential signals, it is often sufficient to route the traces close to each other and make sure that any coupling will appear as a common mode signal
- Obvious guideline
  - Keep digital signals away from sensitive analog nodes
- In SC circuits, most sensitive nodes to watch out for are charge conservation nodes (e.g. op-amp inputs)
  - Any moving node that couples in via parasitic cap will modulate charge and therefore inject noise…

Substrate Types



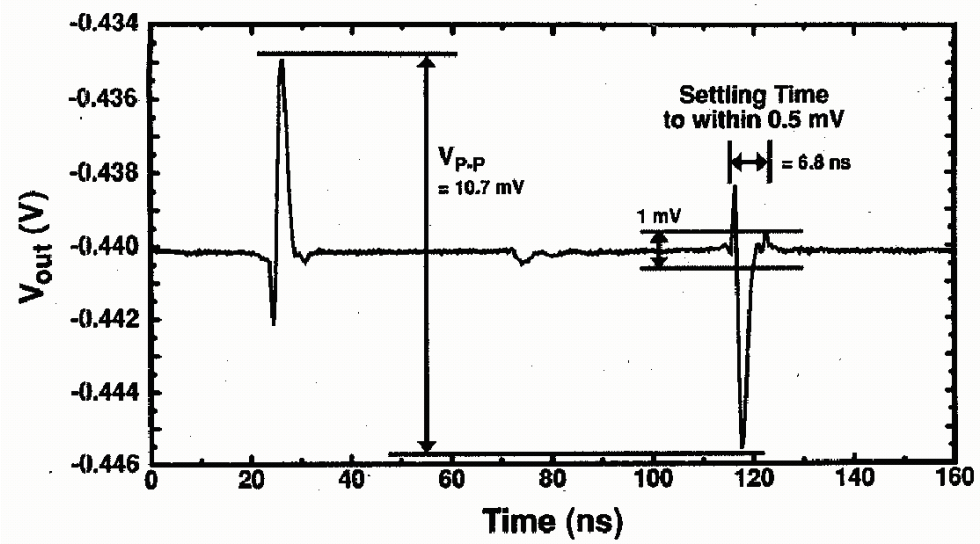
"Epi Substrate"

Epitaxial Substrate



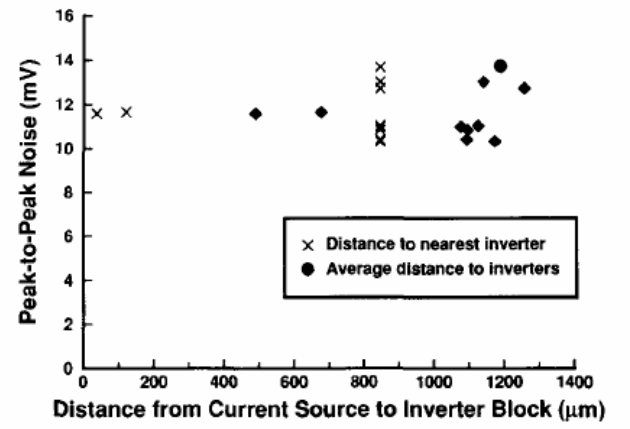
D. K. Su, M. J. Loinaz, S. Masui, and B. A. Wooley, "Experimental results and modeling techniques for substrate noise in mixed-signal integrated circuits," *IEEE Journal of Solid-State Circuits,* vol. 28, pp. 420 - 430, April 1993.

Observed Waveforms



- Current disturbance roughly ± 1%

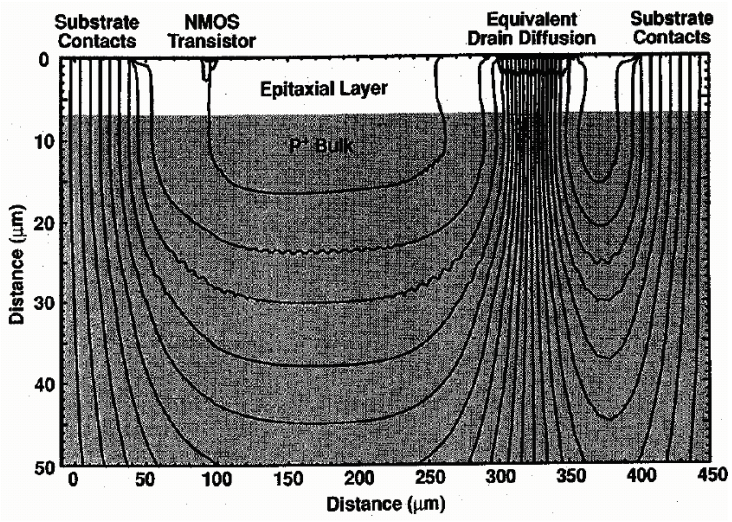
Coupling vs. Distance



- Essentially independent of distance!
  - Why?

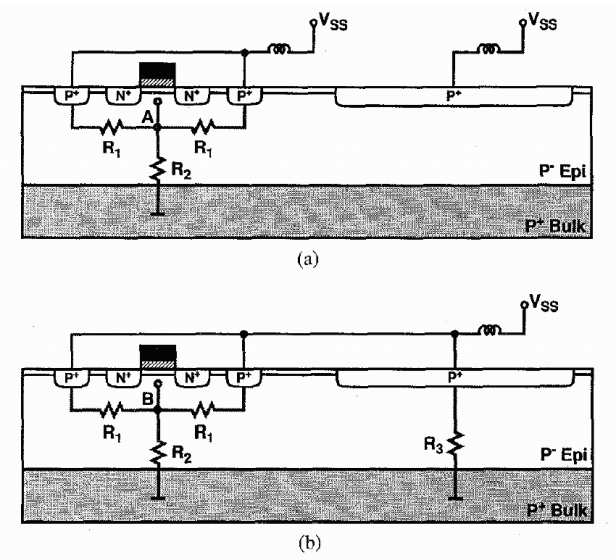
Current Flow in Epi-Substrate

(Setup as in slide 26)

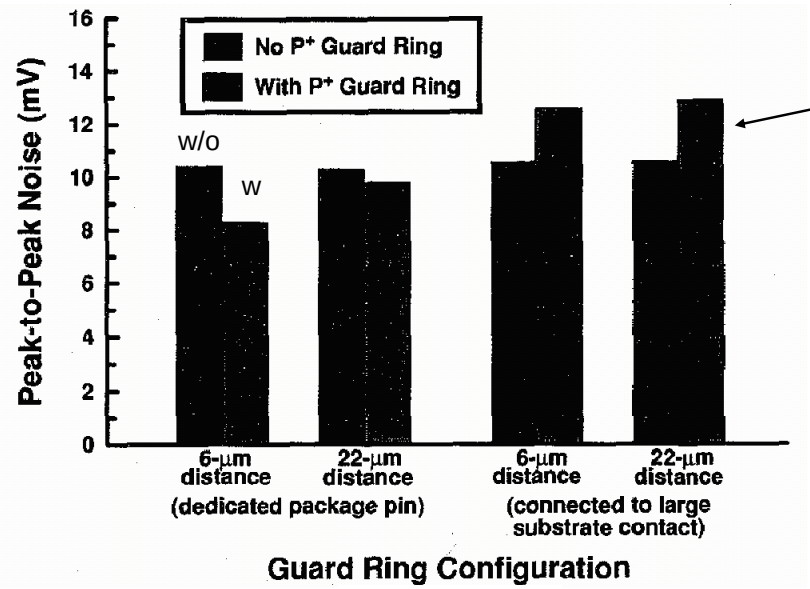


- Majority of current flows in low-resistivity wafer
- Coupling is very weak function of distance

Guard Ring



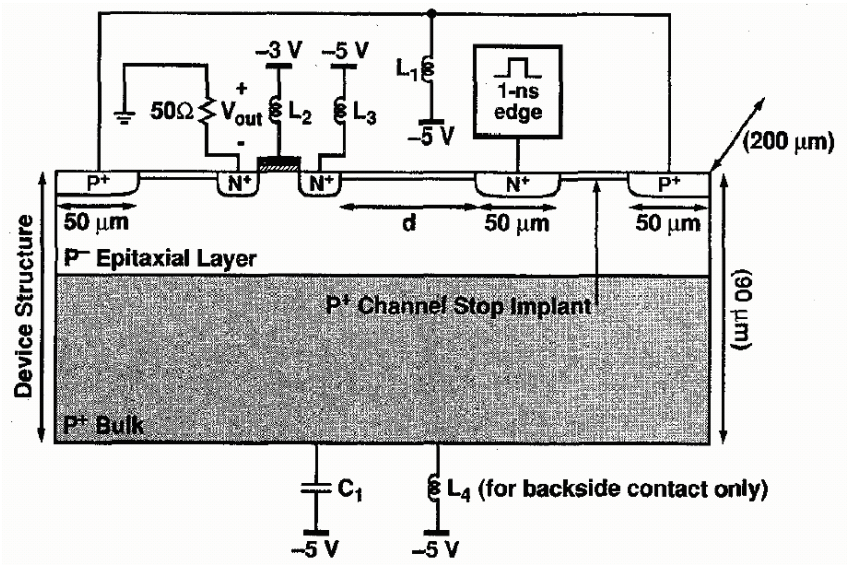
Effect of Guard Ring



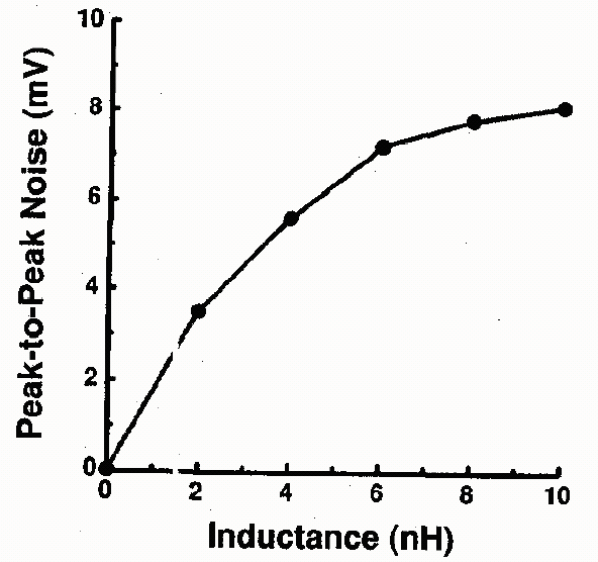
Large guard rings increase coupling!

Epi substrate

Backside Contact



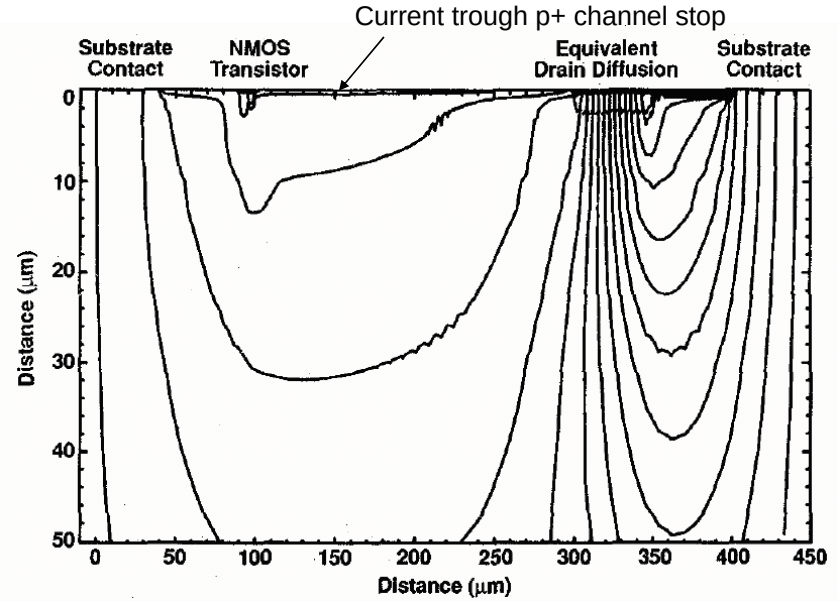
Noise vs. L_4



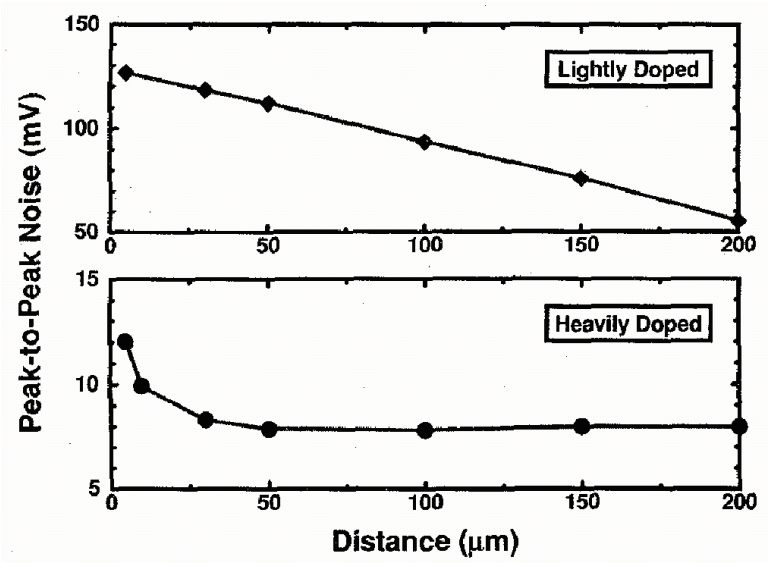
Summary (Epi-Substrate)

- Closely modeled by a "single node"
- The most effective way to reduce coupling in Epi-substrates to is to provide a good, low inductance backside contact
- Unfortunately distance and guard rings don't help much in reducing coupling
- If you decide to use guard rings, make sure to use dedicated guard ring potentials
  - Otherwise guard rings may increase coupling!

Current in High Resistivity Substrate



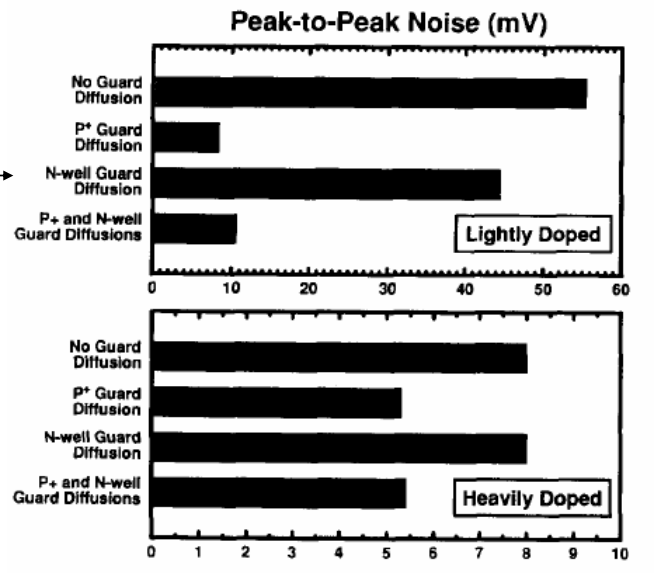
Coupling vs. Distance



(Epi)

Effect of Guard Rings

Breaks p+ channel stop implant →



(Epi)

Summary (Lightly doped substrate)

- Distance and guard rings can help reduce coupling significantly
- Must connect guard rings to quiet, dedicated potentials
  - Otherwise they may inject noise!
- Isolation and coupling effects are highly layout dependent
  - If substrate coupling is critical, the designer should invest a good amount of time to think about potential issues and solutions
- CAD tools?
  - Still being developed/finding commercial use

Additional References

- R. Gharpurey and R. G. Meyer, "Modeling and analysis of substrate coupling in integrated circuits," IEEE Journal of Solid-State Circuits, vol. 31, pp. 344 - 353, March 1996.
- Balsha R. Stanisic, Nishath Verghese, Rob A. Rutenbar, L. Richard Carley, David J. Allstot,"Addressing substrate coupling in mixed-mode ICs: Simulation and power distribution synthesis," IEEE Journal of Solid-State Circuits, vol. 29, pp. 226 - 238, March 1994.
- Kuntal Joardar, "A simple approach to modeling cross-talk in integrated circuits," IEEE Journal of Solid-State Circuits, vol. 29, pp. 1212 - 1219, October 1994.
- Nishath Verghese, David J. Allstot, "Computer-aided design considerations for mixed-signal coupling in RF integrated circuits," IEEE Journal of Solid-State Circuits, vol. 33, pp. 314 - 323, March 1998.
- A. Samavedam, A. Sadate, K. Mayaram, and T. S. Fiez, "A scalable substrate noise coupling model for design of mixed-signal ICs," IEEE Journal of Solid-State Circuits, vol. 35, pp. 895 - 904, June 2000.

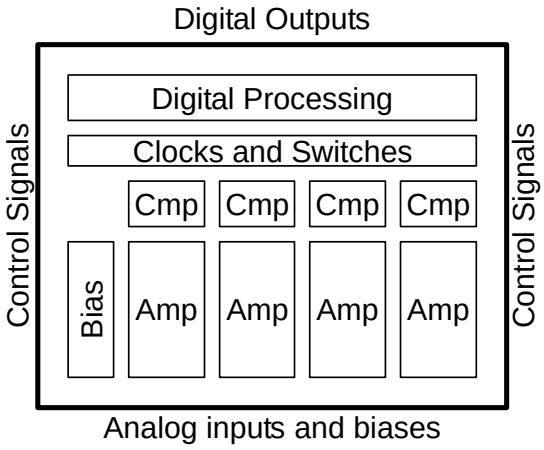
Floorplanning and I/O (1)

- A common mistake is to do a great job of laying out lots of little cells but then make a big mess when pulling the design together
- A good floorplan is essential to being able to quickly make a good layout with few iterations.
- A floorplan is an evolving document that helps the designer organize the chip into pieces that fit together well
  - Don’t be afraid to change it as you go along and discover new issues, just start out with one so you don’t miss the obvious things that can be very painful later.
- When generating a floorplan, keep the ultimate test setup in mind
  - If you have to cross sensitive and noisy signals, it’s best to do it on chip where you only get a few femto Farads of coupling rather than doing it on the board where you will get much more coupling.

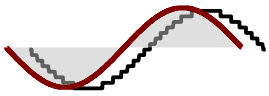
Floorplanning and I/O (2)

- Bond wire and package traces have inductance and resistance. By putting multiple pins in parallel, you can reduce these parasitics.
  - Unfortunately, mutual inductance of neighboring pins fights the reduction. The inductance of two adjacent pins is about 0.7 times that of one, and for three pins, you get about 0.5 times the inductance of one pin.
- Final bit of advice: Know when to stop! You can easily get so carried away with these issues that your layout takes a very long time to complete
  - The key is to do what is right for an application
    - An RF mixer should minimize capacitance
    - A 14-bit A/D converter needs well a very balanced layout
  - Use your own judgment and ask critical questions!

Sample Floorplan (ADC)



Integrated Circuit Filters



Boris Murmann
Stanford University
murmann@stanford.edu

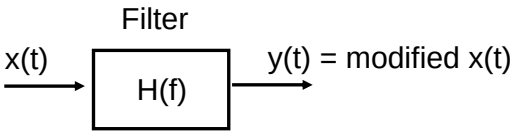
Notes

- Acknowledgement
  - Notes originally compiled by Susan Luschas
  - Edit by Boris Murmann
- References
  - Schaumann, *Design of Analog Filters*, Oxford University Press, 2001.
  - J. Khoury, “Design of a 15-MHz CMOS Continuous-Time Filter with On-Chip Tuning”, IEEE JSSC, Dec. 1991.
  - B. Nauta, “A CMOS Transconductance-C Filter Technique for Very High Frequencies”, IEEE JSSC, Feb. 1992.
  - S. D’Amico, “A 4.1mW 79dB-DR 4th order Source-Follower-Based Continuous-Time Filter for WLAN Receivers”, IEEE JSSC, Dec. 2006.

Outline

- Brief Introduction
- Designing Filter Transfer Functions
- Implementation: Biquad vs. Ladder
- Choosing a Topology
  - Active RC
  - Gm-C
  - Switched capacitor
- Circuit Design Challenges & Examples

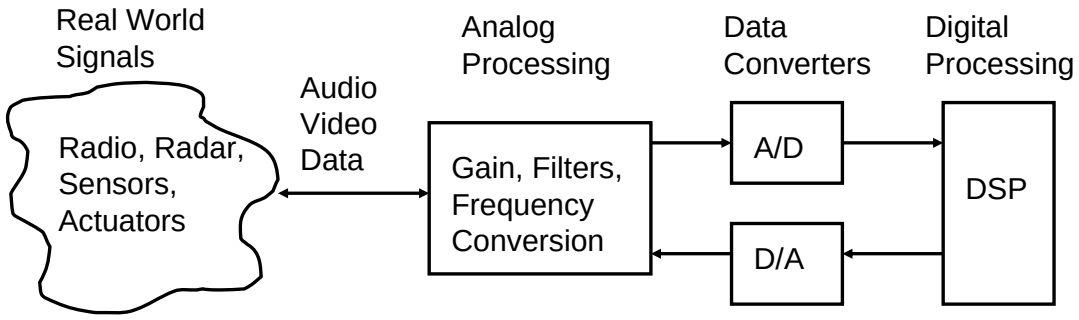
What is a filter?



$$Y(f)=H(f)\,X(f)$$

- Filtering is the process of altering the frequency content of a signal

Why Study Analog & RF Filters?

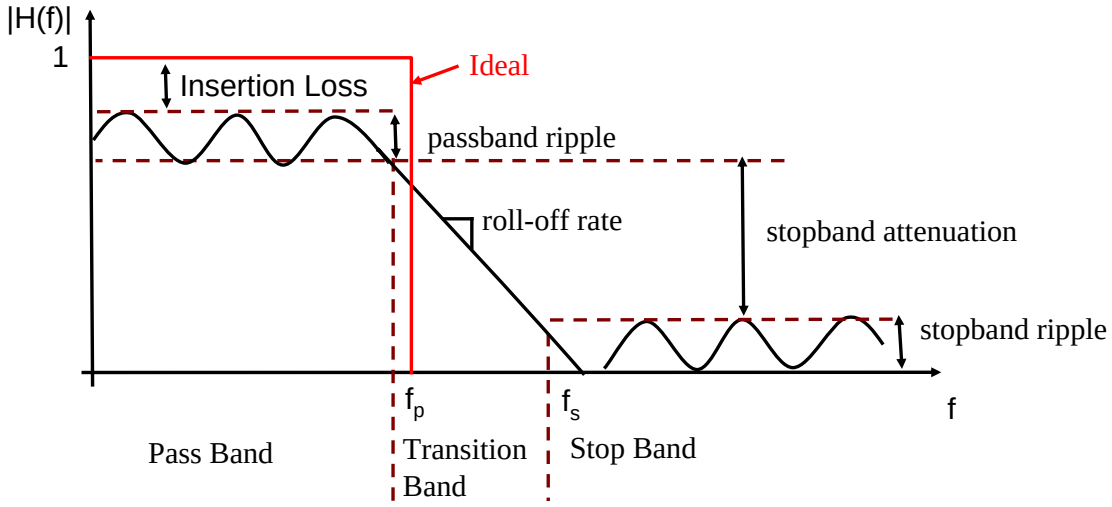


- Data Converters will always be needed to bridge the gap between the digital and real worlds. Filters are needed to band-limit before A/D and re-construct after D/A. Often systems require even more filtering in the Analog Processing block.

Filters are Everywhere

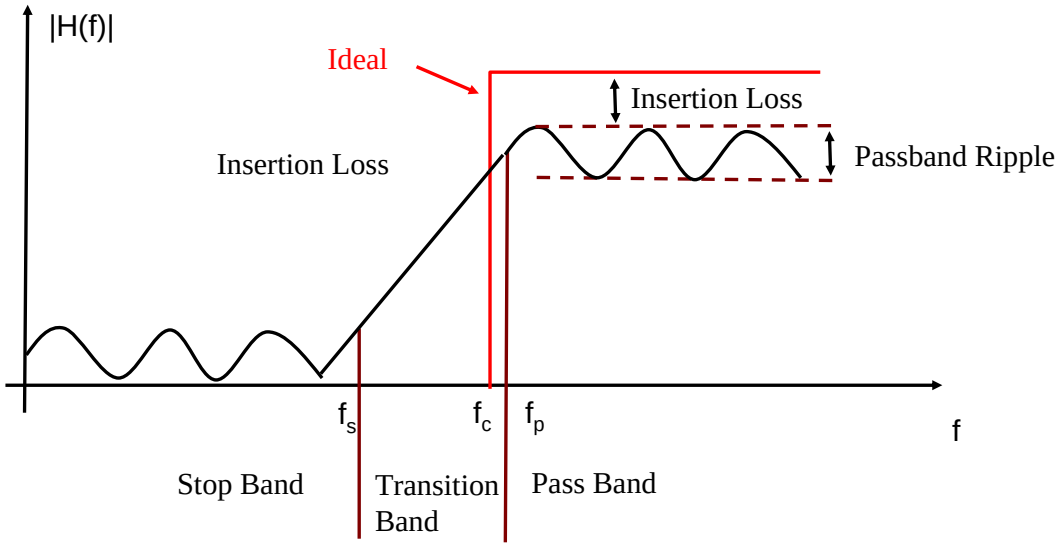
- Audio
  - IPOD, Speakers, Stereo (treble & bass boosting), CD Players, Speech Processing
- Video
  - Cameras, HDTV, XBOX, PSP
- Communications
  - Cell phones, wireless, modems, Ethernet
- Medical, Industrial & Scientific Instrumentation
  - Ultrasound, Radar, pacemakers, hearing aids
- Other
  - Storage Media, Toys, Appliances

Lowpass Filter (LPF)



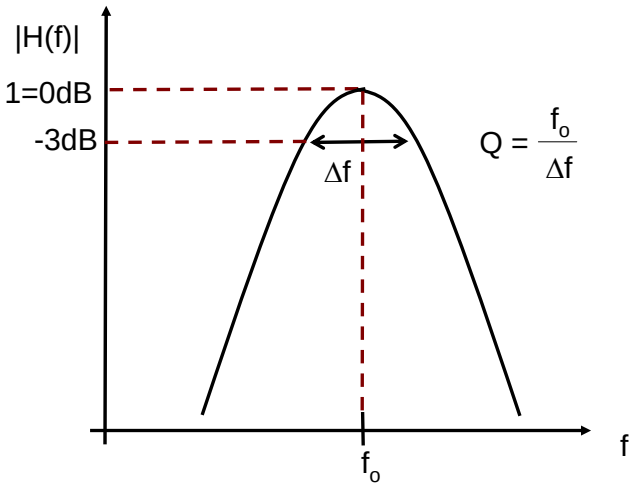
- Typical applications: noise removal, image attenuation, interpolation, amplifier stabilization, data smoothing or averaging

Highpass Filter



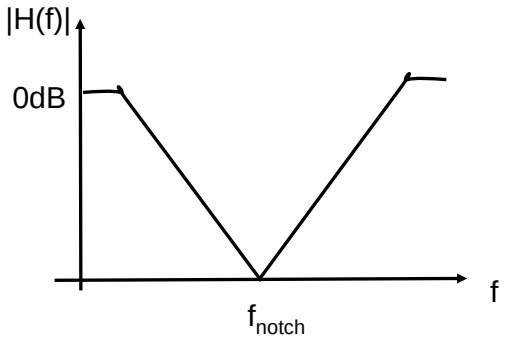
- Typical Applications: DC blocking, edge detection or enhancement

Bandpass Filter (BPF)



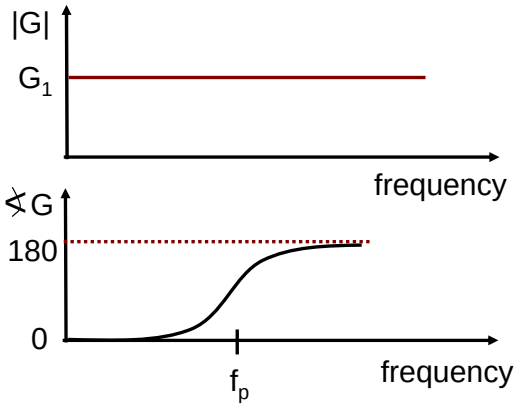
- Bandpass filters are typically used to tune in to a specific channel (radio, TV, etc)

Bandstop, Bandreject, Notch Filter



- Typically used to remove noise at a particular frequency, e.g. 60Hz noise from a power supply.

Allpass Filter



- Allpass filters create a frequency-dependent phase shift.
- Typically used for delay equalization.

Filter Design Goals

- No insertion loss
- Infinite stop-band rejection
- Linear phase
- Low noise, high dynamic range
- Cheap to manufacture, easy to build
- Insensitive to component variations
- Low power

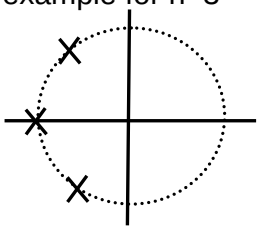
- Want it all, but in reality there are tradeoffs

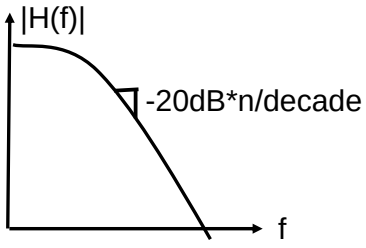
Filter Design Procedure

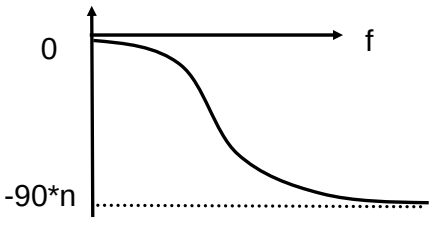
- Determine desired frequency response from system specifications.
- Typically use Matlab or quick hand calculations (if low order filter) to obtain a rational transfer function with left half plane poles that approximates the desired frequency response
- Try to minimize the order of the filter (power & cost). Typically design transfer function with some margin for circuit parameter variations
- Realize the filter: tradeoffs in power, cost, performance determine what kind of realization is chosen
- Build the filter, considering circuit non-idealities

Butterworth Summary

Pole-Zero Plot
example for n=3



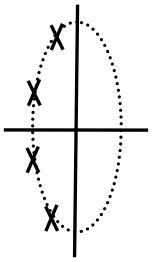


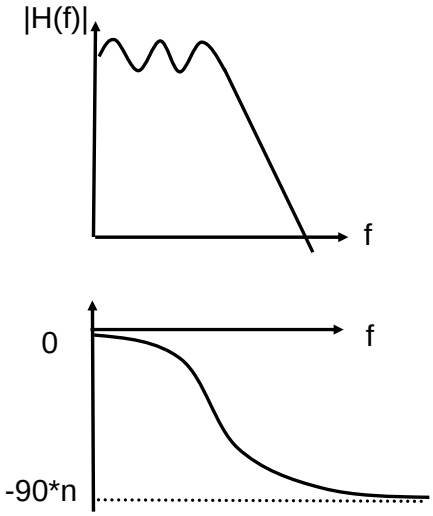


- Poles lie along a circle.
- No ripples ('maximally flat')

Chebyshev (Type I) Filter Summary

Pole-Zero Plot
example for n=4

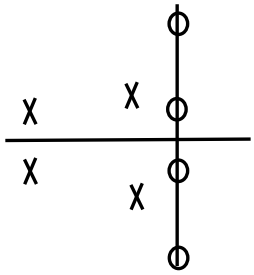


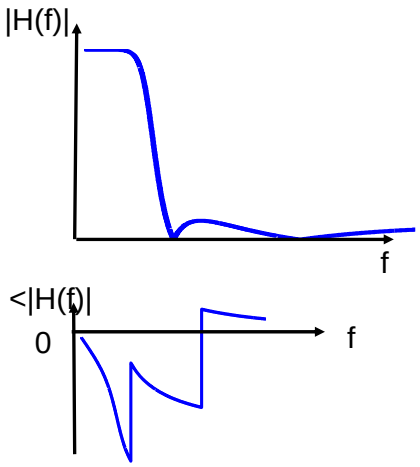


- Poles lie along an ellipse
- ☺ Steeper transition band than Butterworth
- ☺ More stopband attenuation than Butterworth
- ☹ Passband ripples
- ☹ More nonlinear phase than Butterworth

Inverse Chebyshev Summary

Pole-Zero Plot
example for n=4

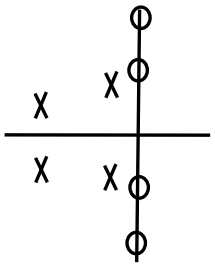


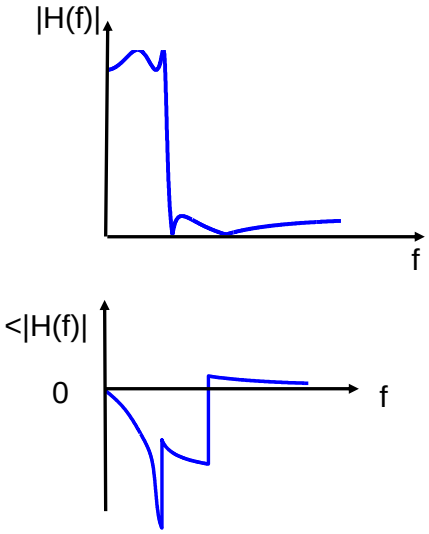


- Ripples in passband traded for ripples in stopband.
- Similar order to Chebyshev for same response.
- Very nonlinear phase response

Elliptic Filters

Pole-Zero Plot
example for n=4

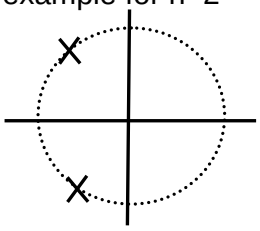


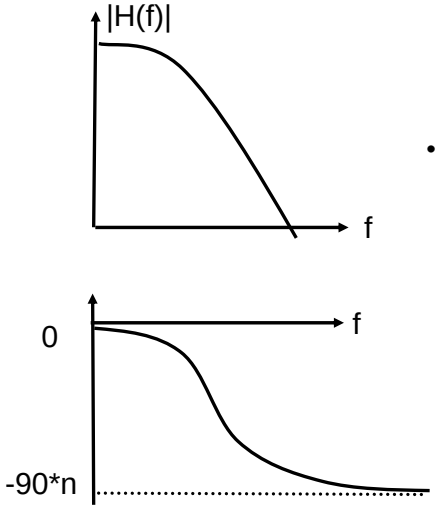


- Ripples in passband and stopband
- Steeper transition band than Chebyshev
- Very nonlinear phase response

Bessel-Thomson Summary

Pole-Zero Plot
example for n=2





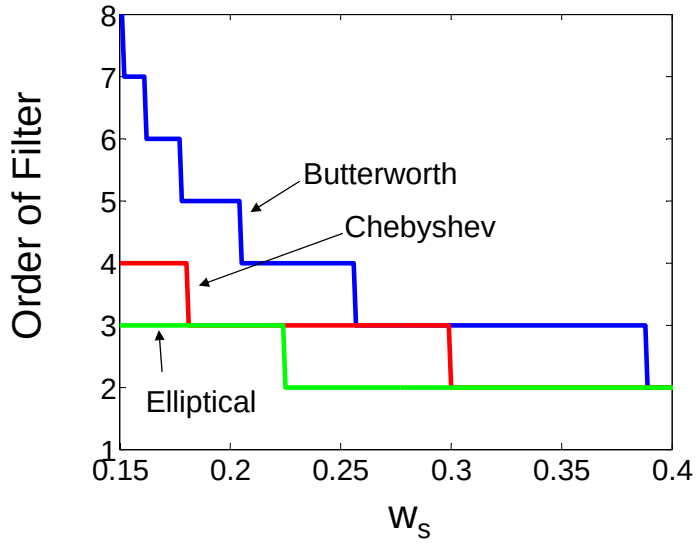
- Trades wider transition band and passband attenuation for maximally flat delay in the passband
- No riginging or overshoot in step response

Comparison of Filters

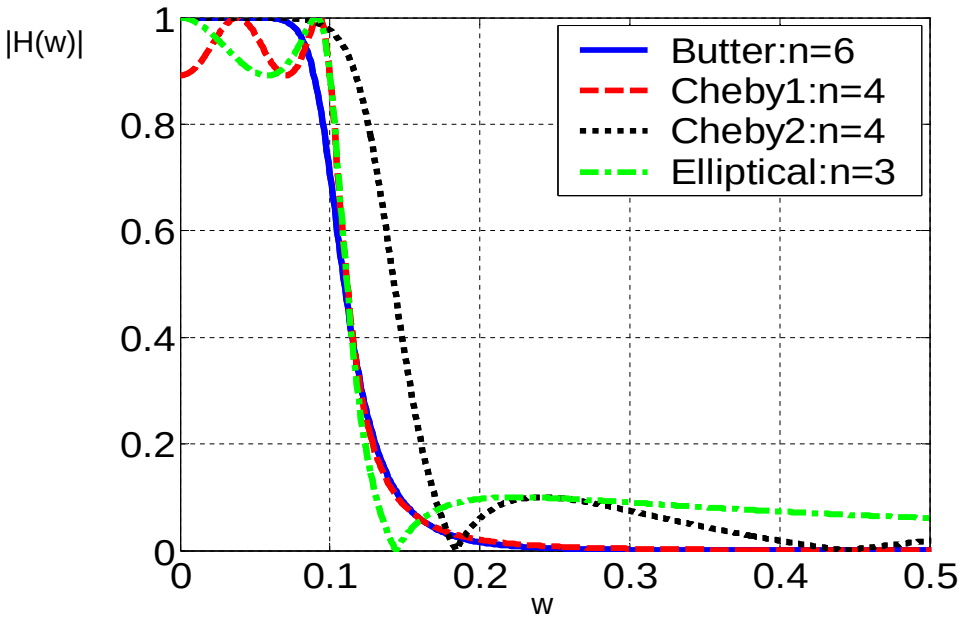
- Comparisons based on
  - Order of filter (cost)
  - Passband response
  - Stopband response
  - Transition band
  - Ease/cost of circuit implementation
- In practice, all are about equally difficult to implement
- Difference in filter order (size and expense of components) matters most

Order Comparison

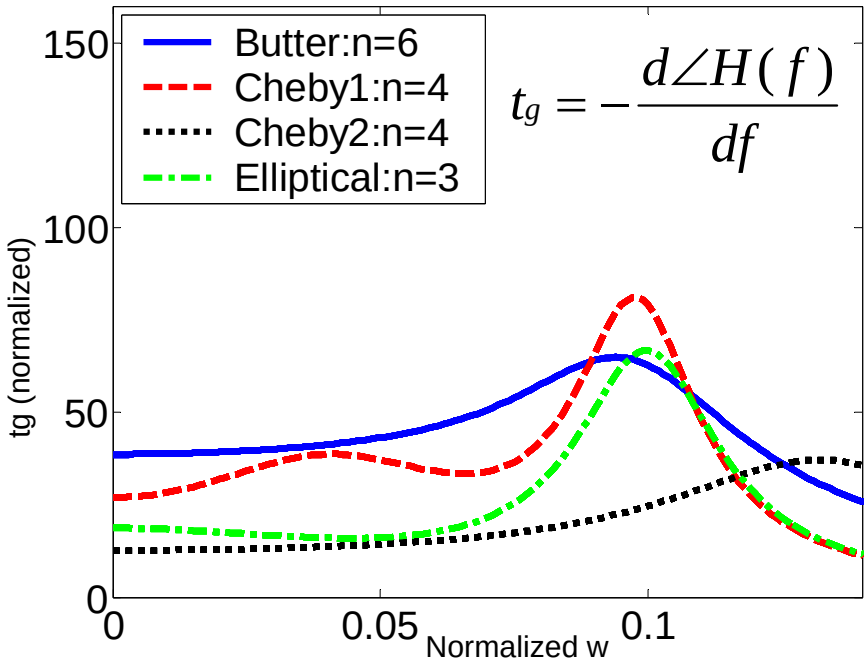
For 1dB ripple in the passpand, 20dB attenuation in the stopband, and w_p=0.1:



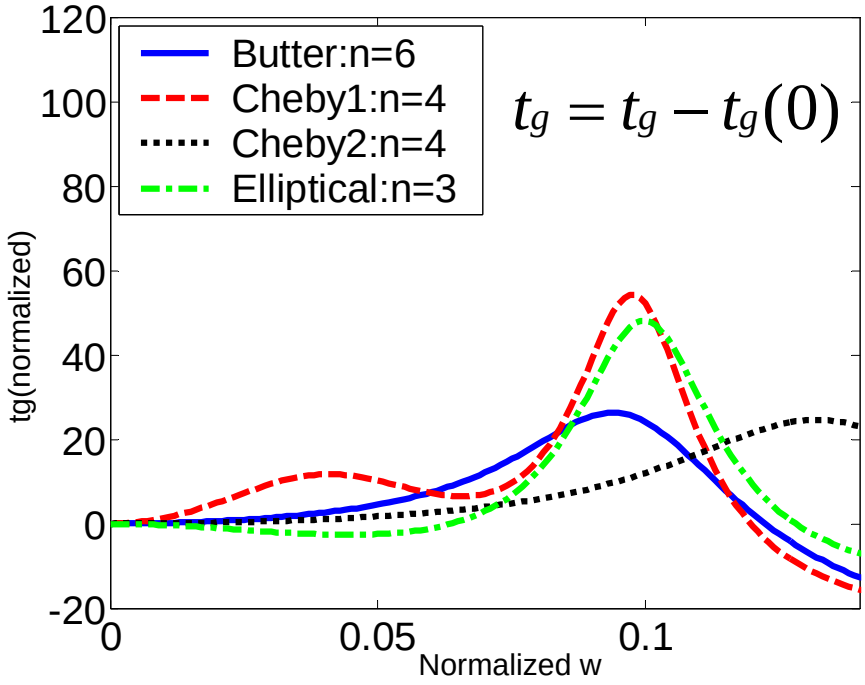
Transfer Function Comparison for ws=0.17



Group Delay



Group Delay Variation



What if we want a High Pass Filter?

Frequency Transformation Summary

If $H_{LPF}(s)$ is a lowpass filter with cutoff frequency $w_c=1$, then…

$H_{HPF}(s)= H_{LPF}(w_c/s)$ w_c=desired cutoff of HPF

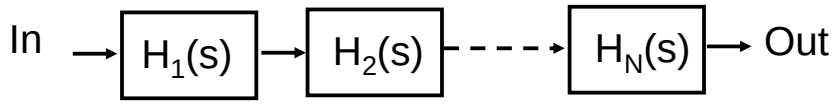
$H_{BPF}(s)=H_{LPF}\left(\frac{s^2+wo^2}{s*B}\right)$ w_o=desired center frequency

B=w_1-w_2=desired width of BPF or notch filter

$H_{notch}(s)=H_{LPF}\left(\frac{s*B}{s^2+wo^2}\right)$

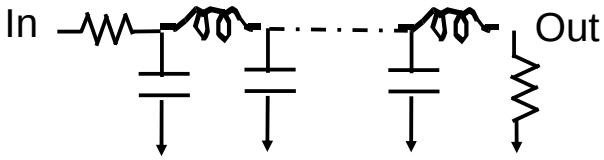
Implementing Filter Transfer Functions

1) Cascade of Biquads

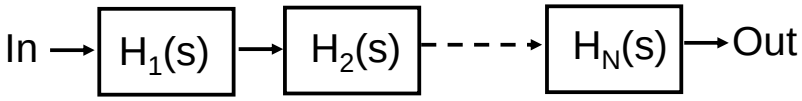


$H(s) = \prod_{i=1}^{N} H_i(s)$, where $H_i(s)$ is a second order transfer function.

2) Ladder

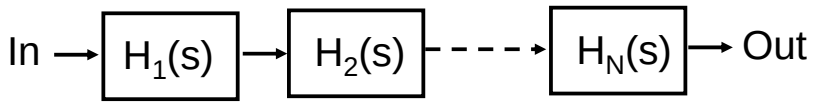


Biquad Cascade Design



- Pole-zero pairing
  - Assign the zero pairs to the closest pole pairs
- Section Ordering
  - lowpass or bandpass biquad as first section to remove large interference
  - In general, choose Q1<Q2<…<QN

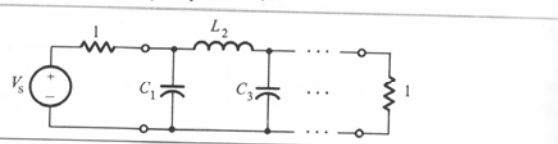
Biquad Cascade Summary



- Advantages
  - Simple Configuration
  - Easy Tuning
- Disadvantages
  - More sensitive to component variations than a ladder

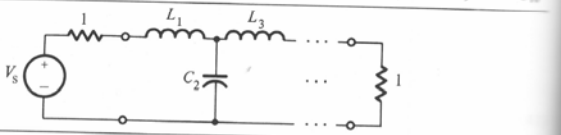
Ladder Design (1)

TABLE 13.1 Table of Element Values for Doubly Terminated Butterworth Filters for $n = 2$ to $n = 10$ Normalized to Half-Power Frequency of 1 rad/s



n	C_1	L_2	C_3	L_4	C_5	L_6	C_7	L_8	C_9	L_{10}
2	1.414	1.414								
3	1.000	2.000	1.000							
4	0.7654	1.848	1.848	0.7654						
5	0.6180	1.618	2.000	1.618	0.6180					
6	0.5176	1.414	1.932	1.932	1.414	0.5176				
7	0.4450	1.247	1.802	2.000	1.802	1.247	0.4450			
8	0.3902	1.111	1.663	1.962	1.962	1.663	1.111	0.3902		
9	0.3473	1.000	1.532	1.879	2.000	1.879	1.532	1.000	0.3473	
10	0.3129	0.9080	1.414	1.782	1.975	1.975	1.782	1.414	0.9080	0.3129

n	L_1	C_2	L_3	C_4	L_5	C_6	L_7	C_8	L_9	C_{10}

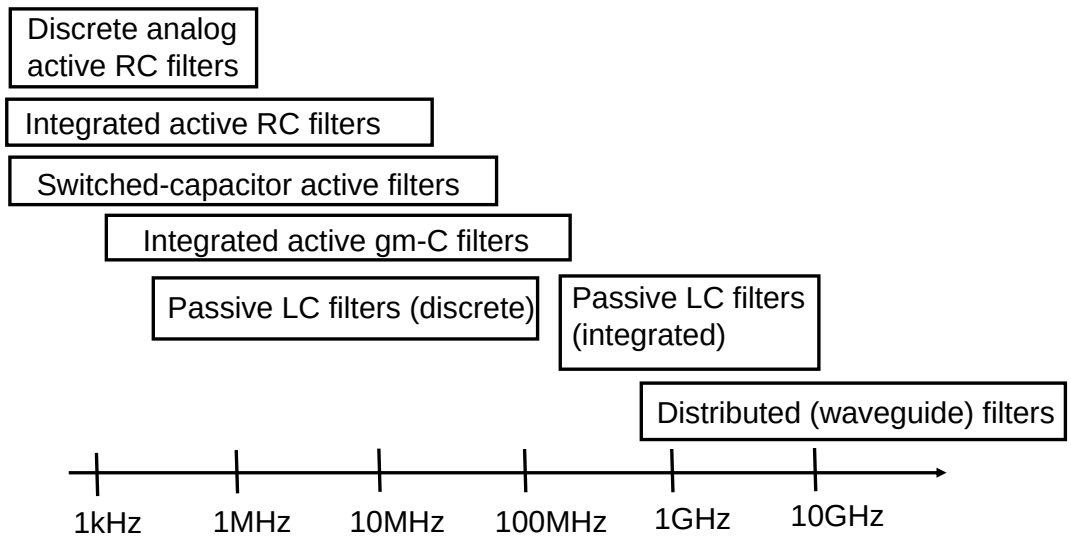


Schaumann, *Design of Analog Filters*, Oxford University Press, 2001.

Ladder Design (2)

- Replace passive elements (usually inductors, resistors) with active components (e.g. gm-C, switched capacitor)
- Often choose topology with minimum inductors, capacitors at input & output so parasitics are absorbed into these capacitances
- Main Advantage: less sensitive to component variations than a biquad implementation
- Without proper resistive termination, the sensitivity will be worse
- With resistive termination, may need a buffer to drive the next stage

Choosing an Implementation



Overview: Topology Tradeoffs

- Opamp RC filters
  - Good linearity, high dynamic range (60-90dB)
  - RC product is difficult to control, typically needs tuning
  - Medium usable signal BW <~10MHz
- Gm-C
  - High frequency performance (>100MHz)
  - Dynamic range not as high as opamp RC (40-70dB)
  - Noise & distortion performance ~60dB
  - Typically needs tuning

Topology Tradeoffs Continued

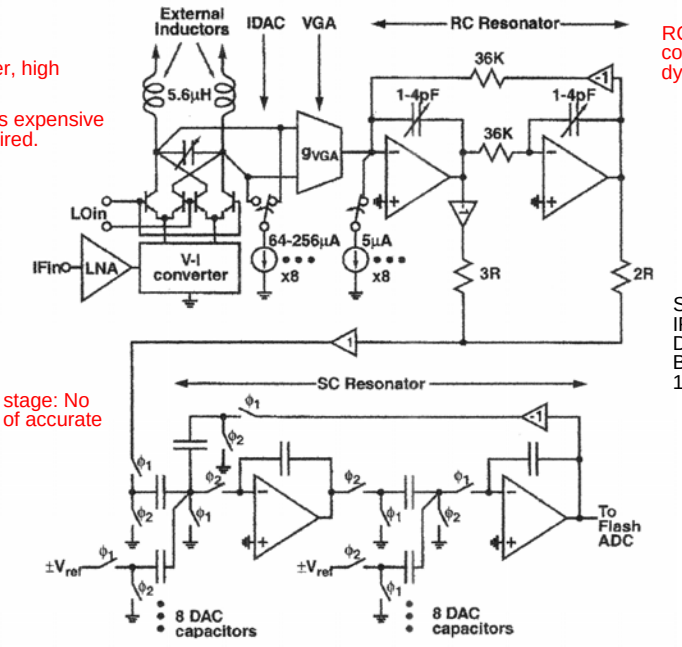
- Switched Capacitor
  - Typically no tuning required (accurate integrated capacitor ratios and accurate clock frequencies)
  - Noise & distortion performance ~ 90dB
  - Frequencies limited by the clock frequency to ~<50MHz
- LC Filters
  - Low power, low noise, high dynamic range (if inductors are ideal)
  - Building inductors is costly (area)
  - Performance degraded by the Q of the inductors

Example: "Neapolitan" Filter

LC first stage:

Low noise & power, high dynamic range

Off-chip inductor is expensive and tuning is required.

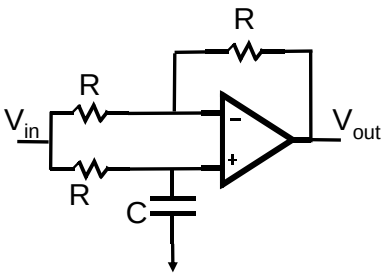


RC second stage: Reduced cost over LC and large dynamic range / low noise

Switched Capacitor final stage: No tuning required because of accurate capacitor ratios.

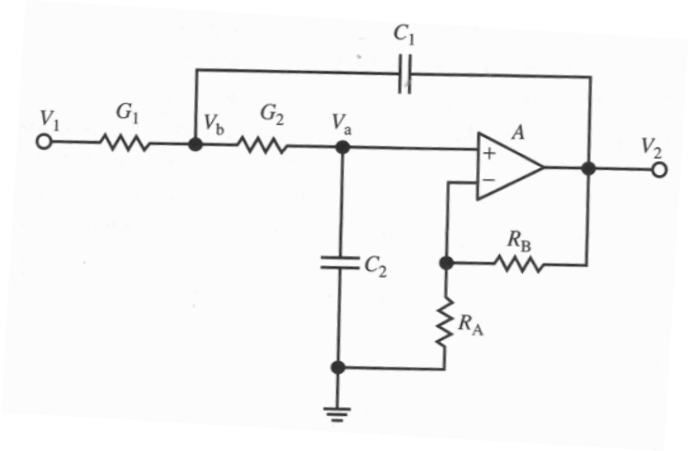
Schreier, et al. "A 10-300MHz IF-Digitizing IC with 90-105dB Dynamic Range and 15-333kHz Bandwidth," *JSSC* Vol. 37 No. 12, December 2002.

Active RC 1st Order Allpass



$$H_{AP}(s) = \frac{V_{out}}{V_{in}} = -\frac{s - 1/RC}{s + 1/RC}$$

Sallen Key 2nd Order Lowpass



$$\omega_0^2 = \frac{G_1 G_2}{C^2}$$

$$Q = \frac{\sqrt{G_1 G_2}}{G_1 + G_2(2-K)}$$

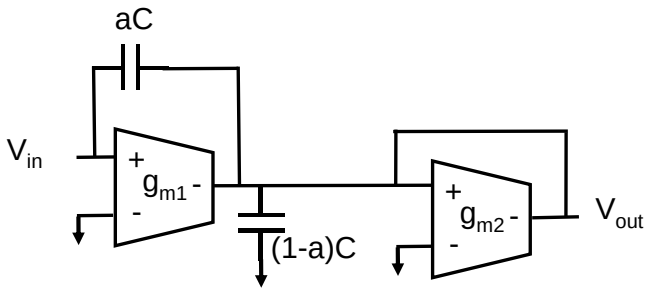
$$H = K > 1$$

$$T(s) = \frac{K G_1 G_2 / C^2}{s^2 + s[G_1 + G_2(2-K)]/C + G_1 G_2 / C^2} = \frac{H\omega_0^2}{s^2 + s\omega_0/Q + \omega_0^2}$$

Active RC Filter Design

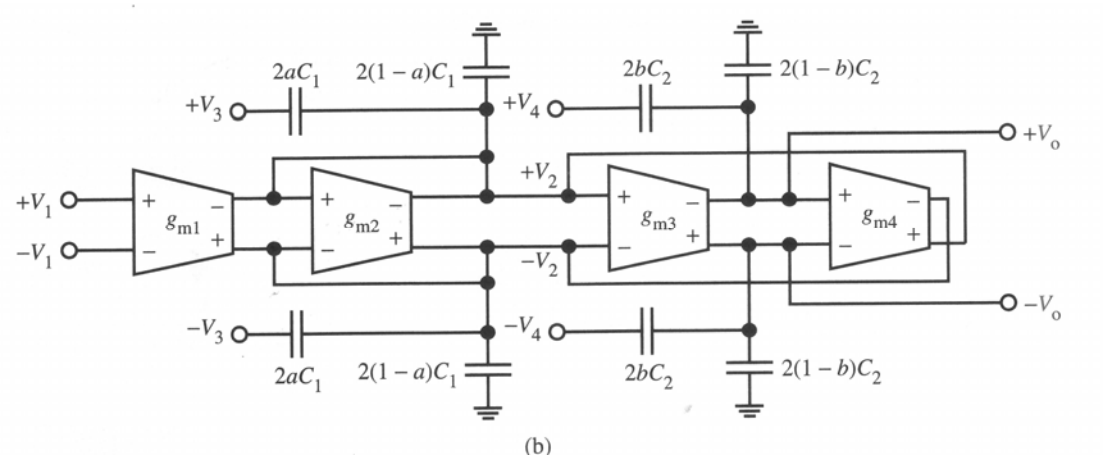
- Finite opamp gain and bandwidth shift the filter poles and create parasitic poles (hopefully out of band)
- Opamp slew rate conflicts with low power design, especially for large loads
- Dynamic range of opamp
- Opamp (and R) noise

Gm-C 1st Order Allpass



$$H_{AP}(s) = \frac{V_{out}}{V_{in}} = \frac{aCs - g_{m1}}{Cs + g_{m2}}$$

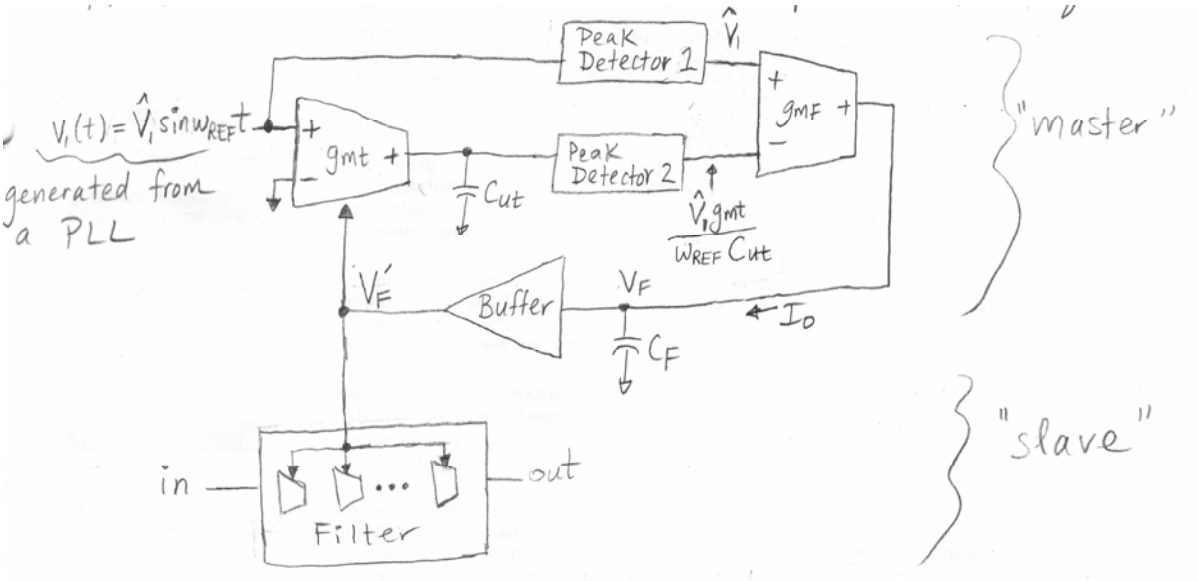
Gm-C 2nd Order Biquad



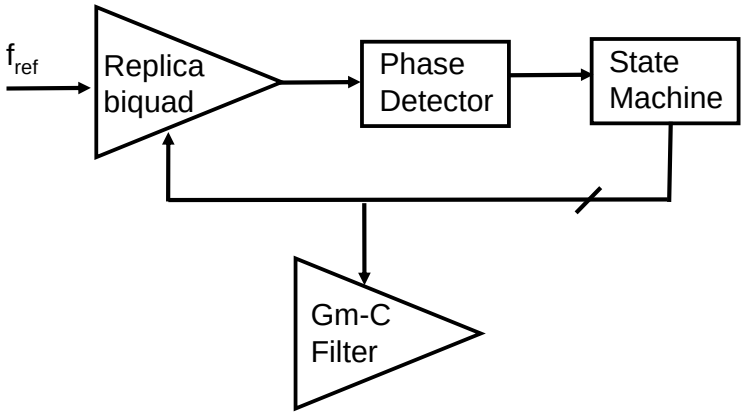
(b)

$$T(s) = \frac{V_o}{V_i} = \frac{s^2 bC_1C_2\frac{V_4}{V_i} + s\left(bC_2 g_{m2}\frac{V_4}{V_i} - aC_1 g_{m3}\frac{V_3}{V_i}\right) + g_{m1}g_{m3}\frac{V_1}{V_i}}{s^2 C_1C_2 + sC_2 g_{m2} + g_{m3}g_{m4}}$$

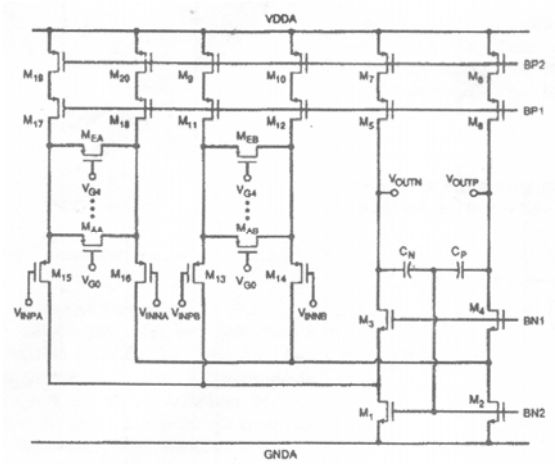
Gm-C Filter Tuning I

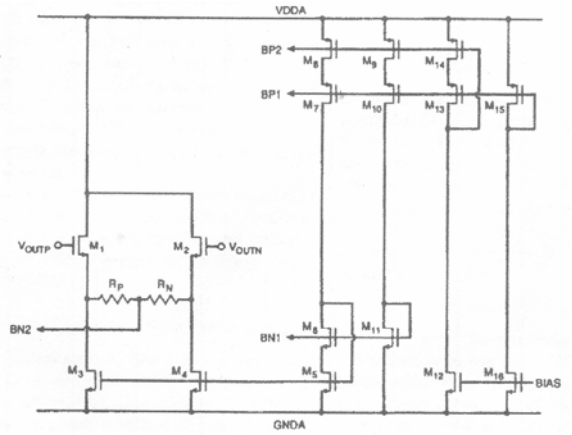


Gm-C Filter Tuning II



Transconductor Implementation I

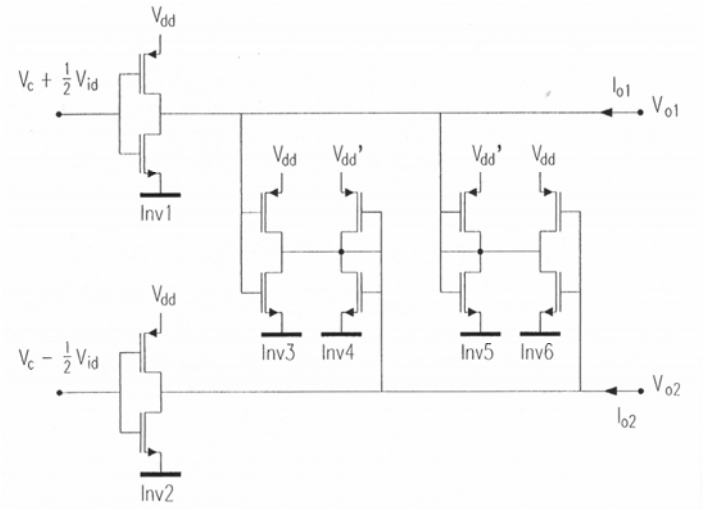




Transconductor Biasing & CMFB

J. Khoury, “Design of a 15-MHz CMOS Continuous-Time Filter with On-Chip Tuning”, JSSC Dec. 1991, p. 1988

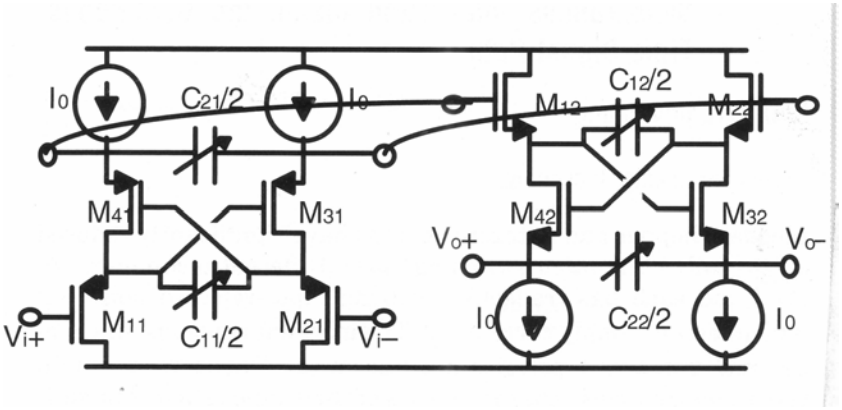
Transconductor Implementation II



Fc=63MHz, Vdd=5V, DR=68dB, CMRR=40dB, 77mW

B. Nauta, “A CMOS Transconductance-C Filter Technique for Very High Frequencies”, JSSC Feb. 1992, p. 142.

Transconductor Implementation III



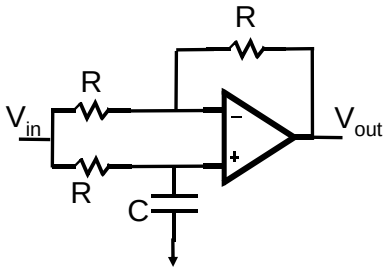
4th order filter, 2.28mA@1.8V , IIP3=17.5dBm

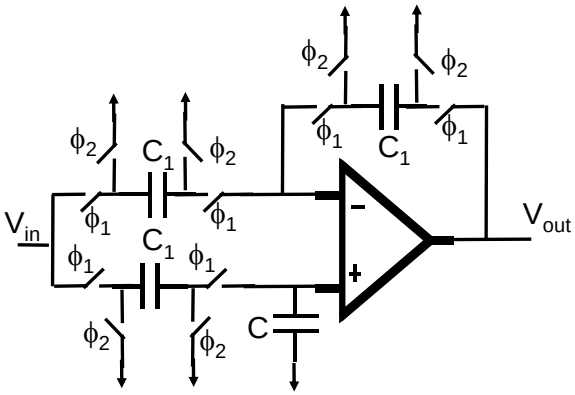
S. D'Amico, "A 4.1mW 79dB-DR 4th order Source-Follower-Based Continuous-Time Filter for WLAN Receivers", ISSC 2006, p. 352.

Gm-C Filter Design Challenges

- Power vs. Linearity
- Dynamic range
- Tuning scheme
- Size of Capacitors / Noise

Switched Capacitor 1st Order Allpass



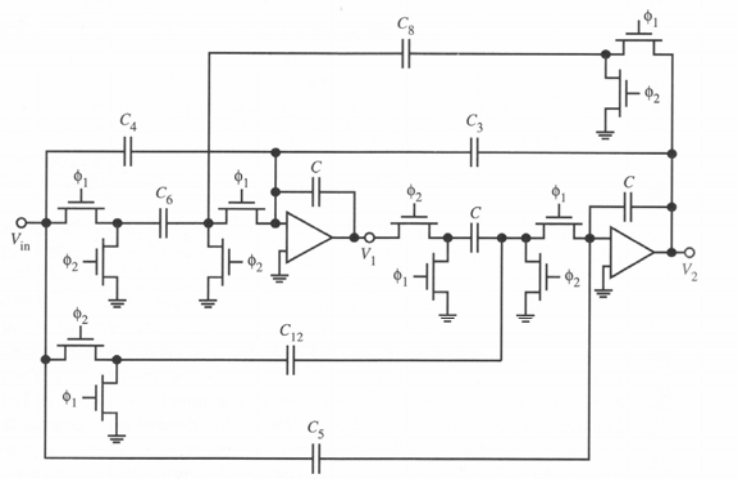


$$H_{AP}(s) = -\frac{s - 1/RC}{s + 1/RC}$$

$$H_{AP}(s) = -\frac{s - C_1 f_{clk}/C}{s + C_1 f_{clk}/C}$$

- Design Strategy: replace R's in RC allpass with switched capacitors.

Switched Capacitor 2nd Order Biquad



$$\frac{V_2}{V_{in}} = -\frac{N(z)}{D(z)} = -\frac{z^2 C_5 + z\left[(C_4 + C_6) - (2C_5 + C_{12})\right] + (C_5 + C_{12}) - C_4}{z^2 C + z\left[(C_3 + C_8) - 2C\right] + (C - C_3)}$$

Designing Switched Capacitor Filters

- Start with active RC filter and replace the R's with switched capacitors
- Use Matlab to design a transfer function in the discrete time domain. Factor the z-domain transfer function and implement as a cascade of integrators, bilinear blocks and biquads
- Starting from a continuous time transfer function, use the bilinear transformation to transform to a z-domain transfer function
- Start with LC ladder prototype and substitute switched capacitor circuits

Non-Idealities in Switched Capacitor Filters

- Opamp noise, kT/C noise
- Finite opamp gain creates gain and phase error
- Capacitor parasitics – use parasitic insensitive switching
- Opamp offset voltage – use correlated double sampling
- Charge injection and clock feedthrough
- Opamp bandwidth and slew rate

Summary

- Filters will be around for a while
- Filter design comes with system level as well as circuit design challenges
- Despite a rich history, new circuit implementations, tuning schemes are still being explored
- This was a brief overview - we've really just scratched the surface!

Additional References **(Haideh Khorramabadi, UC Berkeley)**

- Continuous-Time Filters
  - Y. Tsividis, M. Banu, and J. Khoury, "Continuous-Time MOSFET-C Filters in VLSI", IEEE Journal of Solid State Circuits Vol. SC-21, No.1 Feb. 1986, pp. 15-30 and IEEE Transactions on Circuits and Systems, Vol. CAS-33, No. 2, Feb. 1986, pp. 125-140.
  - Z. Czarnul, "Modification of the Banu-Tsividis Continuous-Time Integrator Structure," IEEE Transactions on Circuits and Systems, Vol. CAS-33, No. 7, pp. 714-716, July 1986.
  - U-K Moon, and B-S Song, "Design of a Low-Distortion 22-kHz Fifth Order Bessel Filter," IEEE Journal of Solid State Circuits, Vol. 28, No. 12, pp. 1254-1264, Dec.1993.
  - H. Khorramabadi and P.R. Gray, "High Frequency CMOS continuous-time filters," IEEE Journal of Solid-State Circuits, Vol.-SC-19, No. 6, pp.939-948, Dec. 1984. K.S. Tan and P.R. Gray, "Fully integrated analog filters using bipolar FET technology," IEEE, J. Solid-State Circuits, vol. SC-13, no.6, pp. 814-821, December 1978.
  - V. Gopinathan, Y. Tsidivis, K-S Tan, R. Hester, "Design Considerations for High-Frequency Continuous-Time Filters and Implementation of an Antialiasing Filter for Digital Video," IEEE Journal of Solid State Circuits, Vol. SC-25, no. 6 pp. 1368-1378, Dec. 1990.
  - A. Durham, J. Hughes, and W. Redman- White, "Circuit Architectures for High Linearity Monolithic Continuous-Time Filtering," IEEE TCAS, pp. 651-657, Sept. 1992.
  - C. Laber and Gray, "A 20MHz 6th Order BiCOM Parasitic Insensitive Continuous Time Filter and Second Order Equalizer Optimized for Disk Drive Read Channels," IEEE Journal of Solid State Circuits, Vol. 28, pp. 462-470, April 1993.
  - H. Khorramabadi, M. Tarsia and N. Woo, "Baseband Filters for IS-95 CDMA Receiver Applications Featuring Digital Automatic Frequency Tuning," 1996 International Solid State Circuits Conference, pp. 172-173.

Additional References

- R. Castello, I. Bietti and F. Svelto, "High-Frequency Analog Filters in Deep-Submicron CMOS Technology", ISSCC Digest of Technical Papers, Feb. 1999, pp.74-75.
- Y. Tsividis, Z. Czarnul and S.C. Fang, “MOS transconductors and integrators with high linearity,” Electronics Letters, vol. 22, pp. 245-246, Feb. 27, 1986.
- I. Mehr and D.R. Welland, “A CMOS Continuous-Time Gm-C Filter for PRML Read Channel Applications at 150 Mb/s and Beyond”, IEEE J. of Solid-State Circuits, Vol.32, No.4, April 1997, pp. 499-513.
- D.A. Johns, K. Martin “Analog Integrated Circuits” Wiley, 1997, p.606
- R. Alini, A. Baschirotto, and R. Castello, “Tunable BiCMOS Continuous-Time Filter for High-Frequency Applications,” IEEE Journal of Solid State Circuits, Vol. 27, No. 12, pp. 1905-1915, Dec. 1992.

• Switched Capacitor Filters
- D. Senderowicz et. al, “A Family of Differential NMOS Analog Circuits for PCM Codec Filter Chip,” IEEE Journal of Solid-State Circuits, Vol.-SC-17, No. 6, pp.1014-1023, Dec. 1982.
- Tat C. Choi, "High-Frequency CMOS Switched-Capacitor Filters," U. C. Berkeley, Department of Electrical Engineering, Ph.D. Thesis, May 1983 (ERL Memorandum No. UCB/ERL M83/31).
- B.S. Song, P.R. Gray "Switched-Capacitor High-Q Bandpass Filters for IF Applications," IEEE Journal of Solid State Circuits, Vol. 21, No. 6, pp. 924-933, Dec. 1986.
- R. Gregorian, G. Temes, “Analog CMOS Integrated Circuits," Wiley, 1986, pp. 277.
- K. Martin, A. Sedra, “Effect of the Opamp Finite Gain & Bandwidth on the Performance of Switched-Capacitor Filters," IEEE TCAS, vol. CAS- 28, no. 8, pp. 822-829, Aug 1981.
- K.L. Lee, “Low Distortion Switched-Capacitor Filters," U. C. Berkeley, Department of Electrical Engineering, Ph.D. Thesis, Feb. 1986 (ERL Memorandum No. UCB/ERL M86/12).
- K. Martin and A. S. Sedra, “Strays-insensitive switched-capacitor filters based on the bilinear z transform,” Electron. Lett., vol. 19, pp. 365-6, June 1979.