Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

Digitization and data transmission

Soumyajit Mandal

EDIT 2023

@BrookhavenLab

Overview

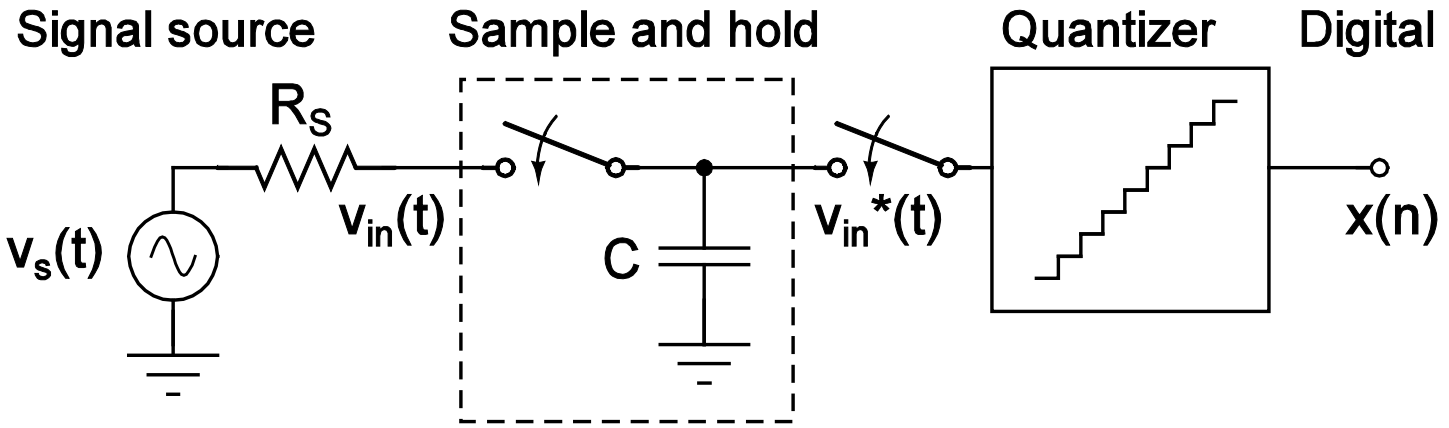
- Analog-to-digital conversion
- Digital data transmission

Analog-to-digital conversion

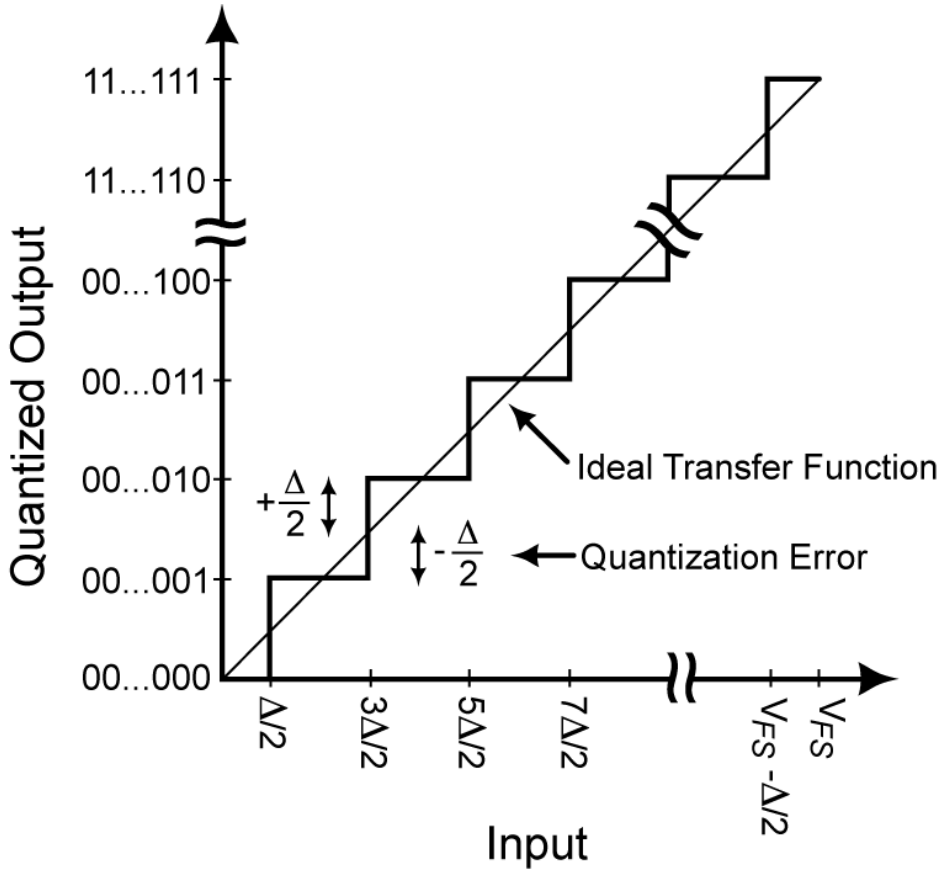
Soumyajit Mandal

Analog-to-digital converters (ADCs)

- The output of an AFE is a continuous, time-varying signal that needs to be converted into a digital number sampled at discrete time steps. This process is called analog-to-digital conversion.
- Analog-to-digital converters sample and then quantize or round-off a real signal to the nearest integer.
- The finer the precision of the converter and the faster the rate of sampling, the better is the approximation of the real signal with a set of numbers that vary with time.
- Before sampling, there is usually an *implicit or explicit* anti-aliasing filter that removes high-frequency content in the signal that can be misrepresented after sampling as false lower-frequency information.
- The Nyquist rate of sampling f_s for a signal "perfectly bandlimited" to f_0 is $f_s = 2f_0$ and is the minimum rate at which the sampling can be performed to yield no error. Perfectly bandlimited signals don't exist since there are no perfectly sharp filters.

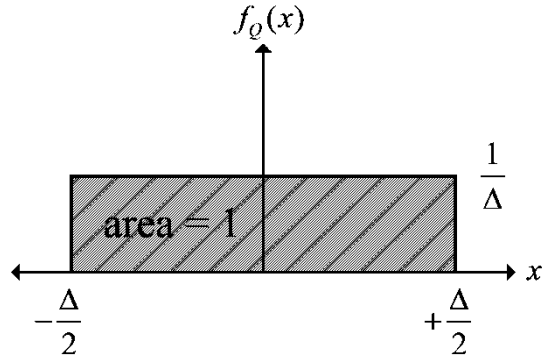


Basics of quantization noise in A/D converters

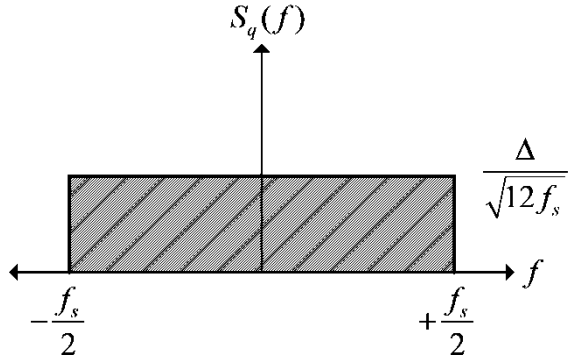


Static performance

Deviations from a linear transfer function quantified by integral and differential nonlinearity (INL and DNL) metrics.



Probability density function of quantization error.

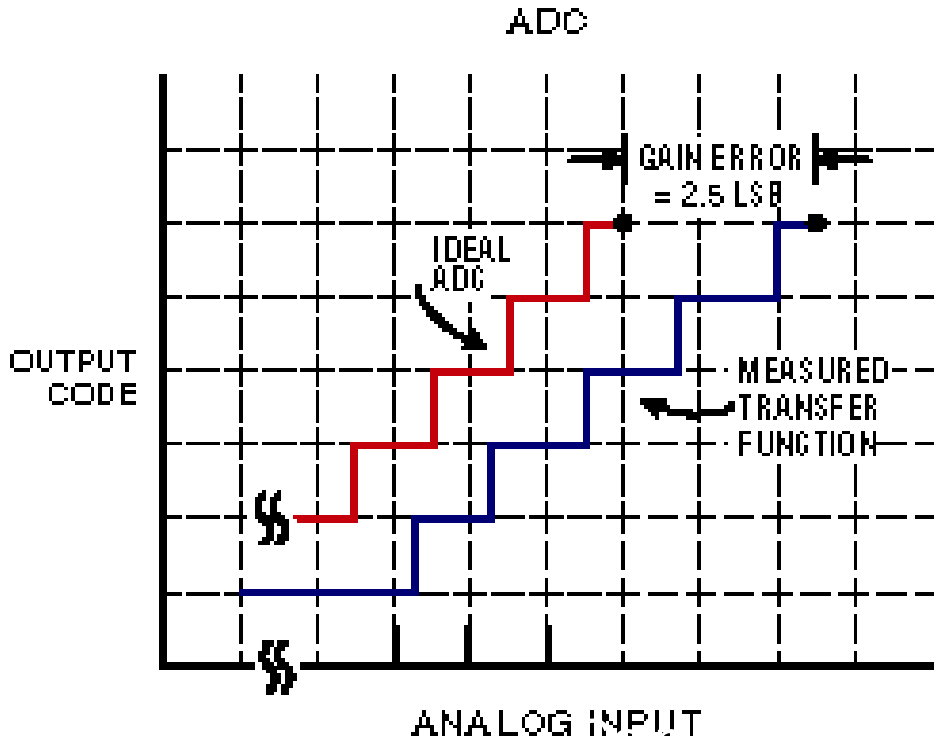


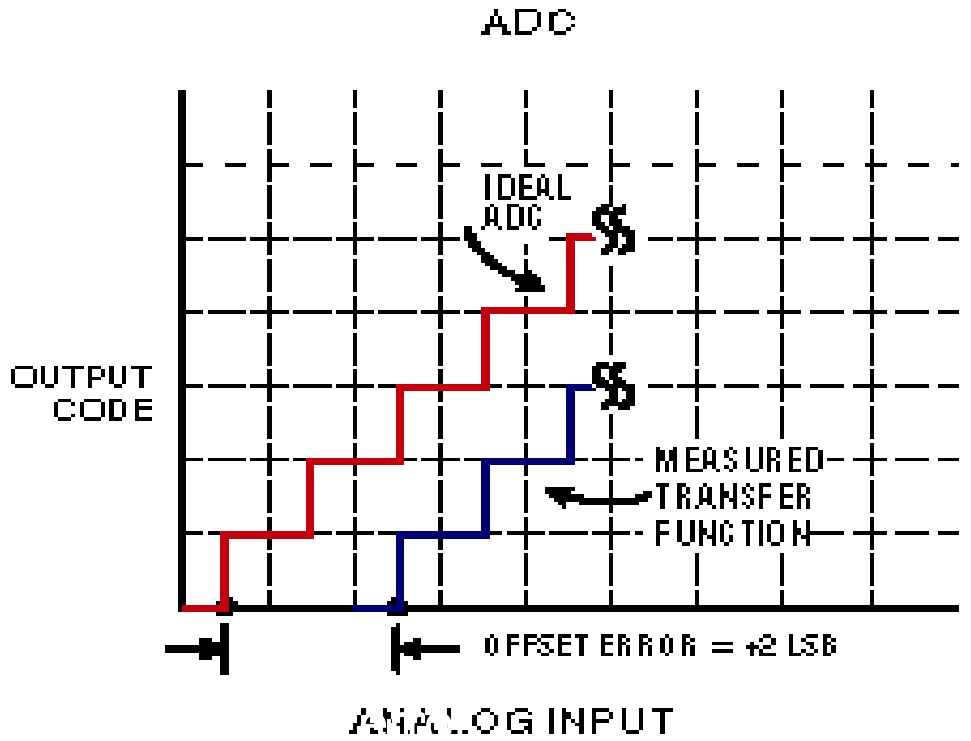
Spectral density of quantization error can be approximated as white noise.

RMS quantization error $\sigma_q = \frac{\Delta}{\sqrt{12}}$.

Brookhaven National Laboratory

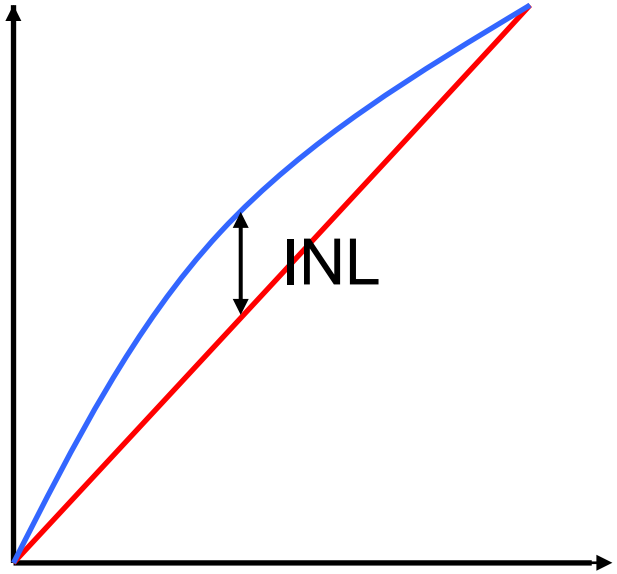
Gain and offset errors



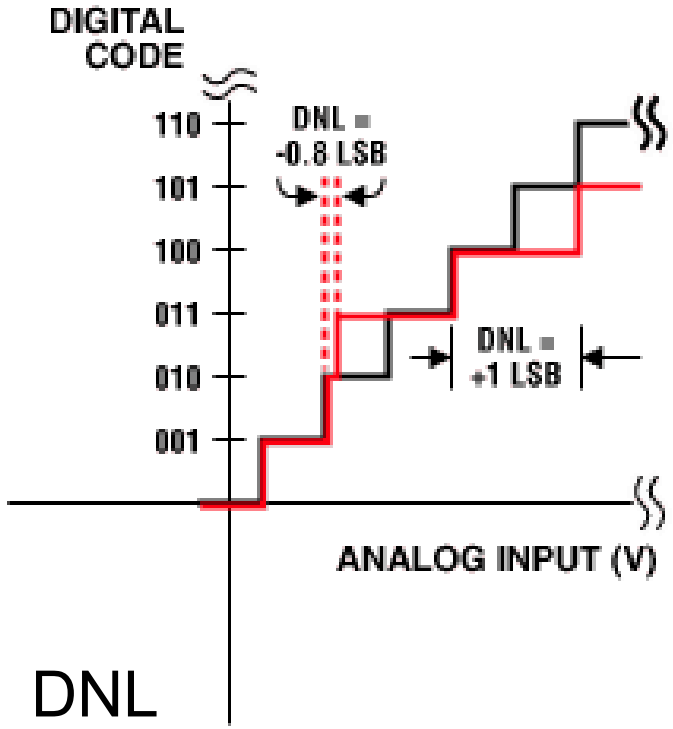


- Gain error measures the deviation of the ADC input-output slope from its ideal value.
- Offset error measures how much input voltage is needed before the ADC produces non-zero output.
- Both gain and offset errors are predictable and can be calibrated out.

INL (integral nonlinearity) and DNL (differential nonlinearity)



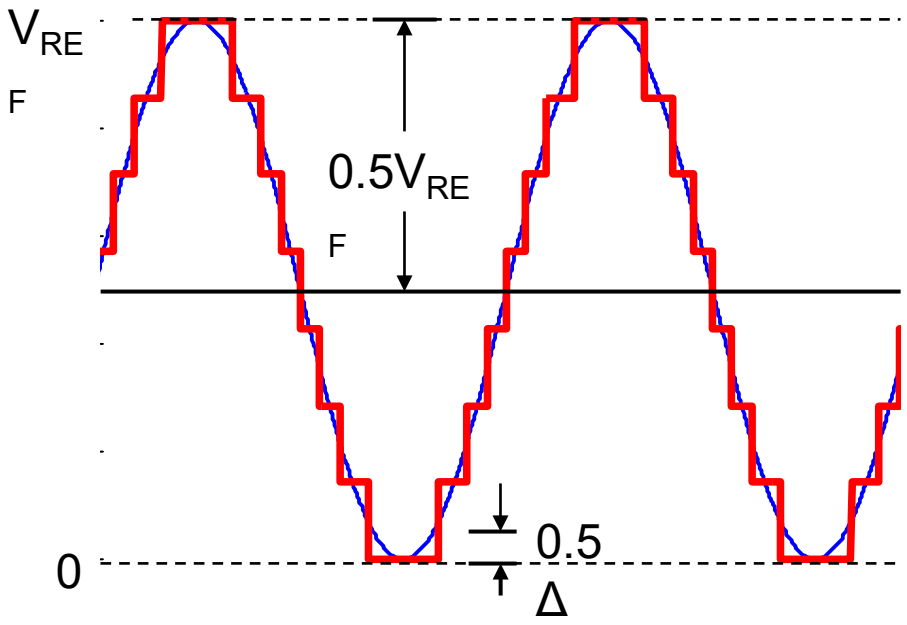
INL



DNL

- Any deviation from 1 LSB input voltage difference between adjacent digital codes constitutes DNL.
- INL is the integral of DNL over the full scale range.
- INL measures the deviation of the ADC input-output characteristic from linearity after gain and offset errors have been removed.
- Neither INL nor DNL can be easily removed from the measurement.

Equivalent number of bits (ENOB)



3 –bit ADC example

$$SNR = \frac{\frac{1}{2}\left(\frac{V_{REF}}{2}\right)^2}{\frac{\Delta^2}{12}}$$

← Signal power

← Quantization noise

$$\Delta = \frac{V_{REF}}{2^{ENOB}}$$

$$\Rightarrow \boxed{ENOB = \frac{SNR_{dB} - 1.76}{6}}.$$

Setup for measuring ENOB

- Feed in a full-scale sinusoid at approximately 0.1 x (sampling frequency).
- Store several cycles of the digital output.
- Window the samples and perform a FFT.
- Measure the resultant SNR from the FFT spectrum and use the formula to get ENOB.

Figure of merit (FOM) for energy-efficient A/D conversion

Dynamic performance

Characterized by effective number of bits (ENOB).

Consider an ideal ADC with a full-scale sinusoidal input of amplitude = $V_{FS}/2$:

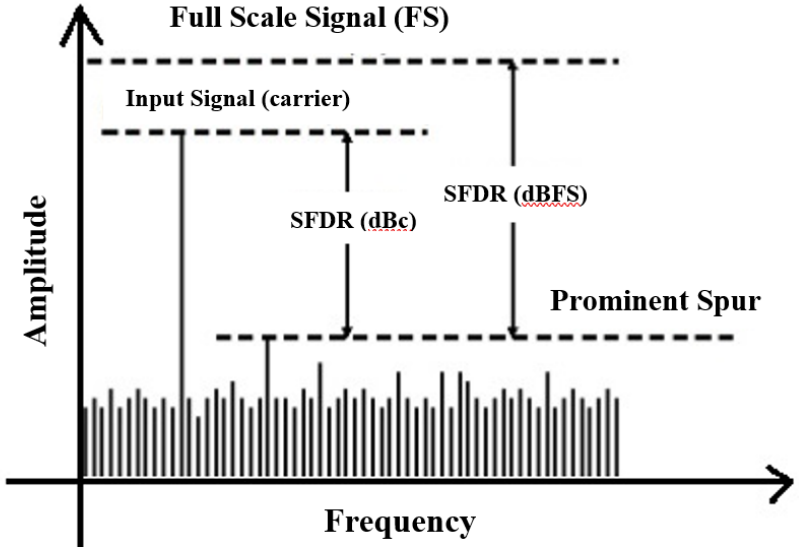
$$SNR_{max} = \frac{P_{sig}}{\sigma_q^2} = \frac{V_{FS}^2/8}{\left(V_{FS}/2^N\right)^2/12} = \frac{1.5}{2^{2N}} = \left(6.02N + 1.76\right)\text{dB}.$$

A real ADC will also add thermal noise and distortion to the output, so the signal-to-noise-and-distortion ratio $SNDR < SNR_{max}$. This defines a more realistic $ENOB < N$:

$$SNDR = \left(6.02 \times ENOB + 1.76\right)\text{dB},$$

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

Another metric: Spurious-free dynamic range (*SFDR*)



$$FOM = \frac{P_{ADC}}{f_S 2^{ENOB}}.$$

- *FOM* is the energy per quantization level (pJ or fJ).
- In general, we want this to be ***small.***

Note: *ENOB* may be dominated by either quantization noise or thermal noise. In thermal-noise-limited converters $2^{2(ENOB)}$ is more appropriate, but most people use 2^{ENOB} for historical reasons.

Brookhaven National Laboratory

Types of analog-to-digital converters

Nyquist-rate converters
- Flash
- Dual-slope
- Successive approximation (SAR)
- Algorithmic
- Pipelined

Oversampled converters
- Sigma-delta (Σ-Δ)

Hybrid converters
- Combine two or more of the basic types (e.g., noise-shaped SAR)

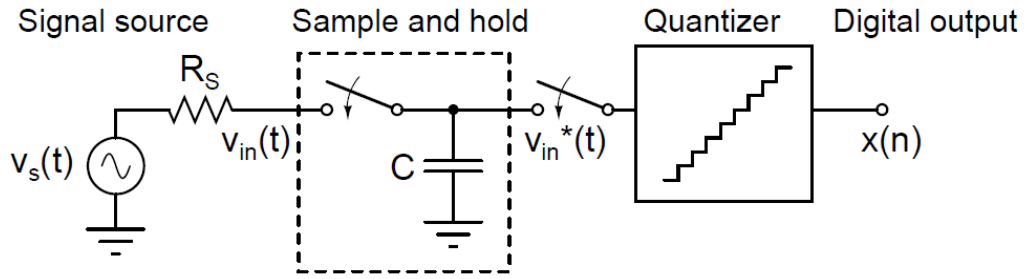
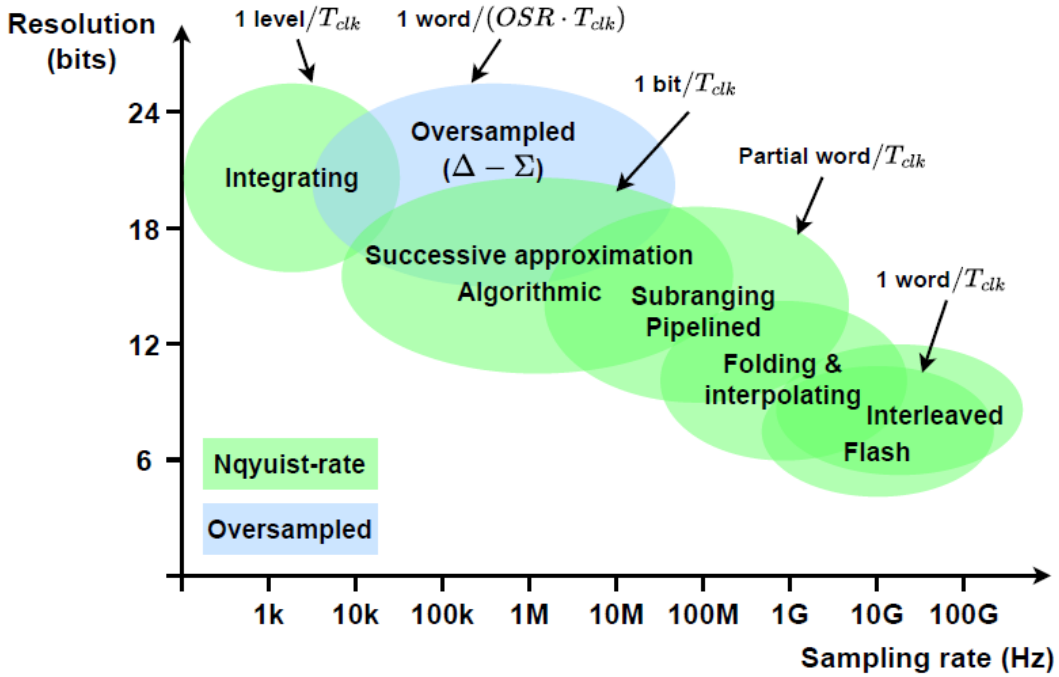
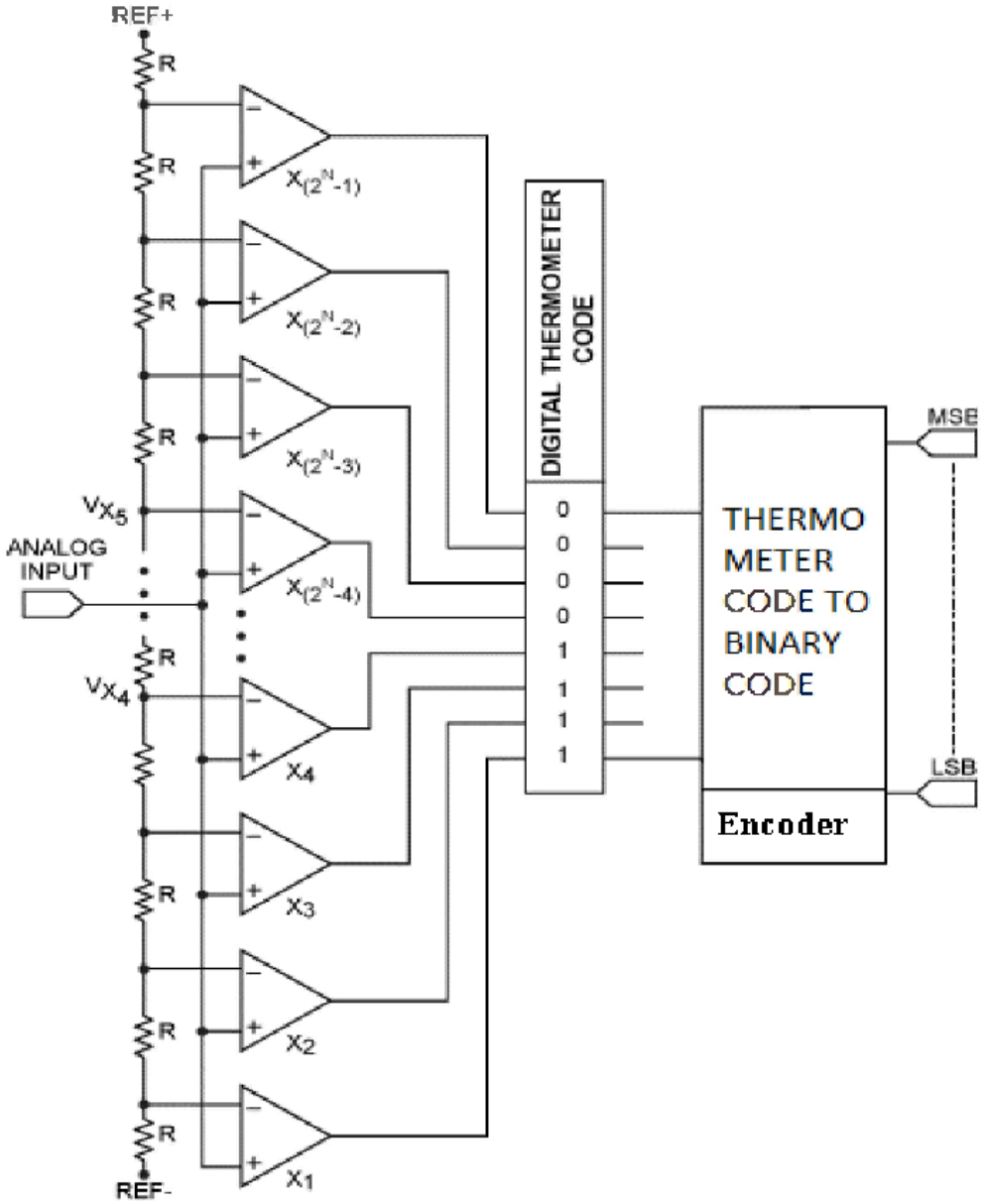


Figure 17.1: Block diagram of a generic ADC.

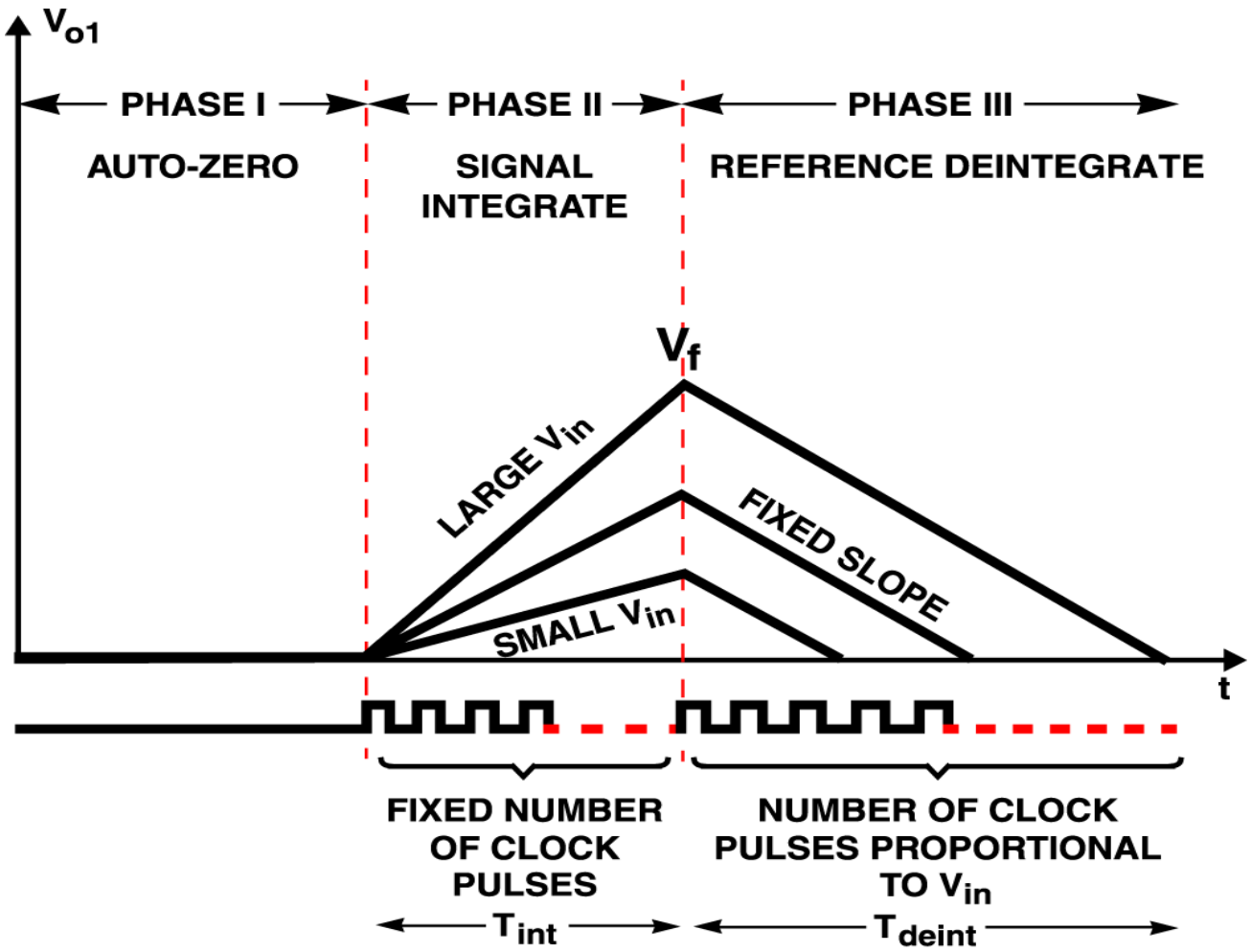


Simplest ADC: The flash converter



- Fastest topology, converts in a single clock cycle.
- Requires 2^N comparators for N bits of resolution. The large number of comparators burn a lot of power and consume a lot of area.
- Requires calibration for precise operation.
- Typically used when raw speed is critical and high resolution is not required, such as in the front-ends of digital oscilloscopes.
- Single-bit flash converters (i.e., single comparators) are used to detect pulses.

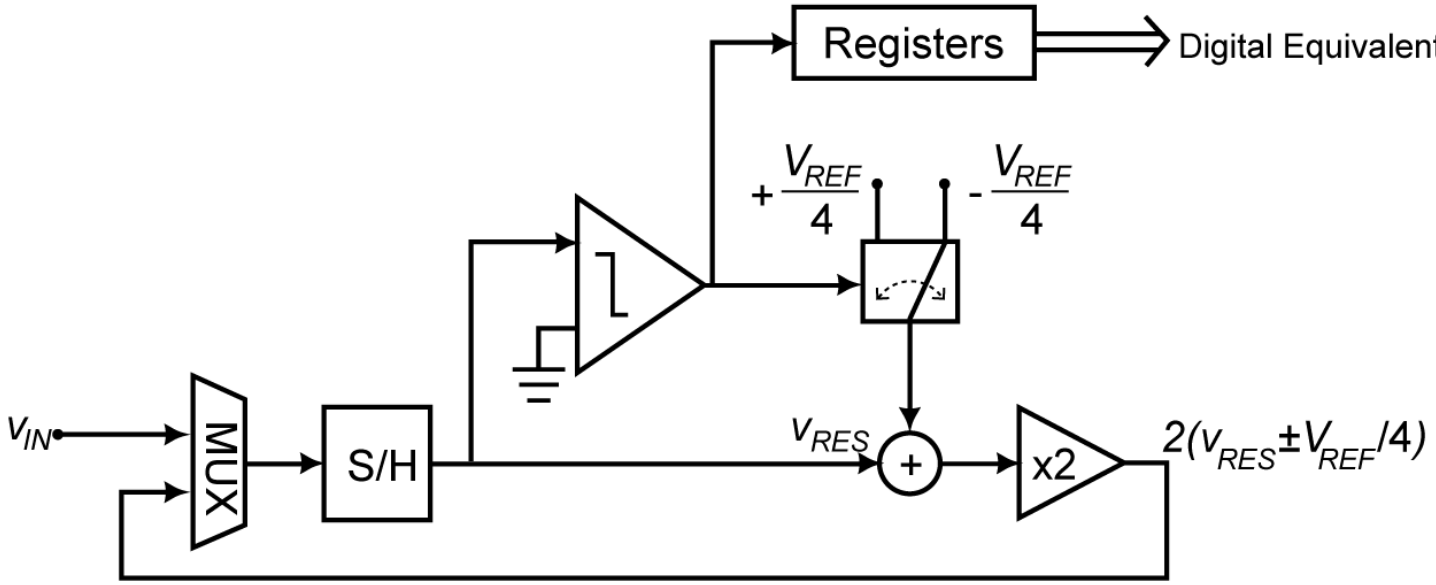
The dual-slope converter

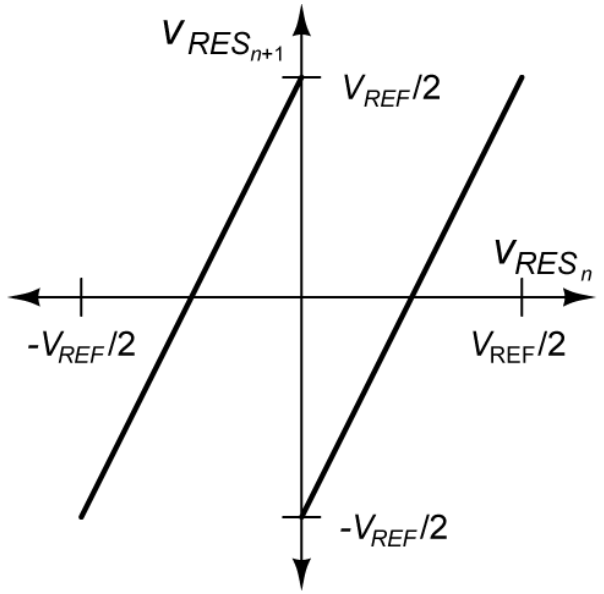


- Also known as an integrating ADC.
- Slow topology, requires 2^N clock cycles for N bits of precision.
- Dual-slope operation cancels offsets and slope errors in the integrator.
- Used in precise but slow applications, e.g., high-precision digital multimeters.
- Single-slope variant, known as a Wilkinson ADC, is often used for off-line digitization of pulses.
  - Can also be used for time interval measurements, i.e., as a time-to-digital converter (TDC), by converting time to voltage.

$$I = C\frac{dV}{dt}, \quad V = \frac{1}{C}\int Idt$$

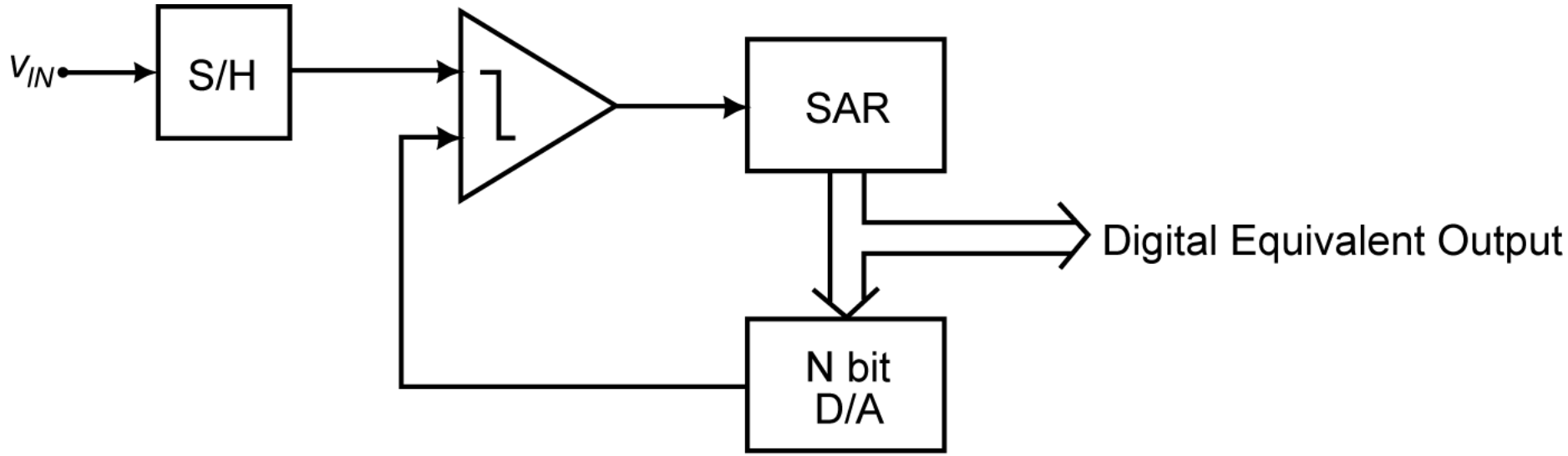
Algorithmic A/D converters





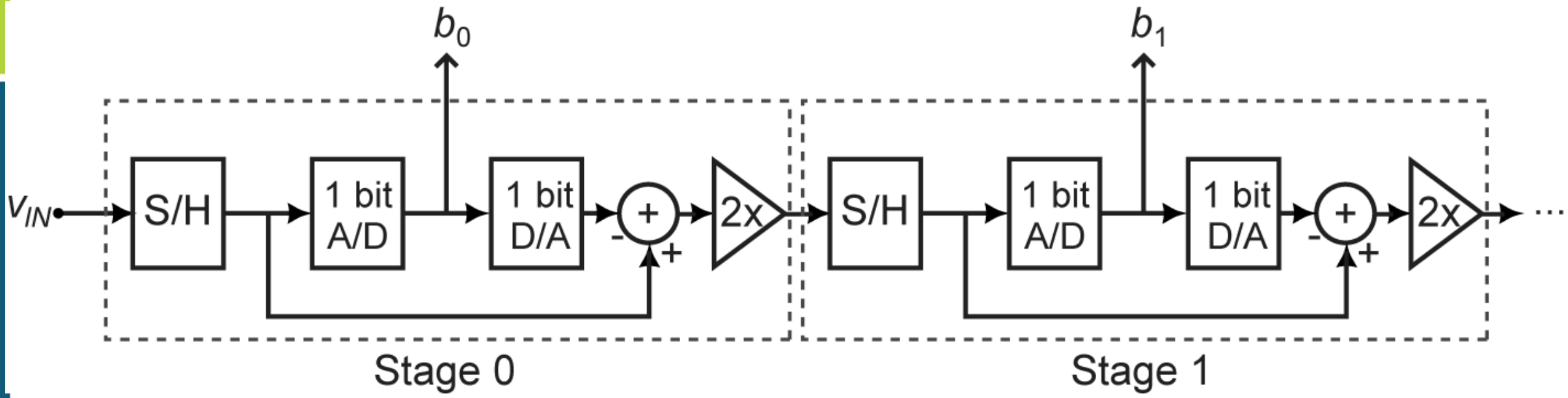
- Inputs within (0, $V_{REF}/2$) or within ($-V_{REF}/2$,0), i.e. in the top or bottom half of the input range, are recursively remapped to ($-V_{REF}/2$, $V_{REF}/2$), i.e. to the full range, in each successive iteration of the conversion. The input is quantized to finer and finer precision as the zoom factor for the residue increases by 2 in each stage of the conversion.
- Also known as a cyclic converter: It uses a single comparator, amplifier, and S/H in each iteration cycle to obtain 1 bit/cycle.
- The residue amplification step is a key source of error.

Successive-approximation converters



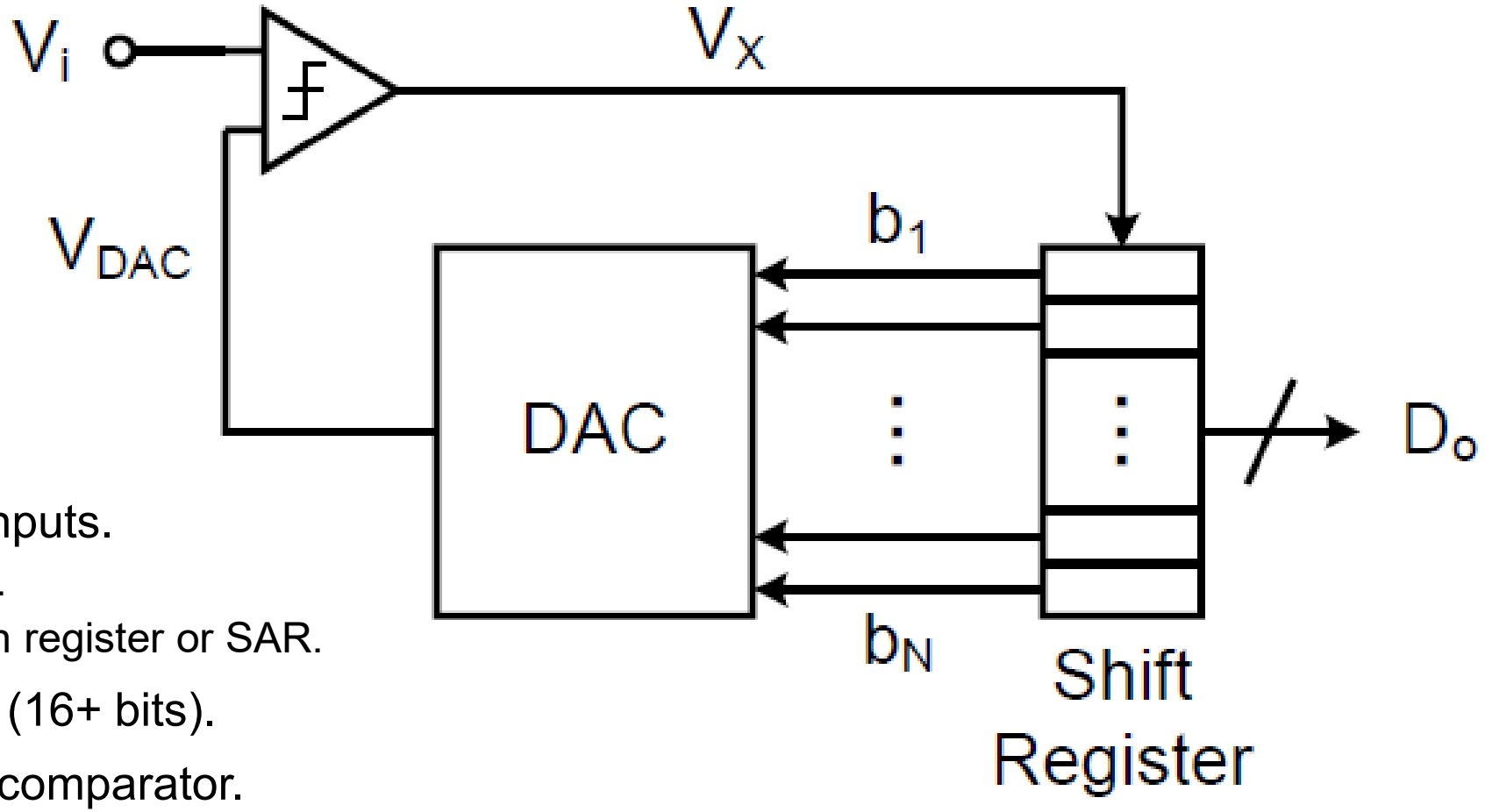
- The architecture uses a S/H, single comparator, successive approximation register (SAR), and an *N*-bit D/A converter.
- The algorithm implements a binary search by successively setting bits of the D/A from MSB to LSB to '1' in each iteration and checking if the best quantized approximation to the input so far is above it or below it; if it's below, the last bit is set to '1', if not, it is set to '0'. The process then moves onto the next bit. The complete digital output appears in *N* clock cycles.
- Highly energy-efficient design since it needs no amplifier and scales with technology.

Pipelined A/D converters



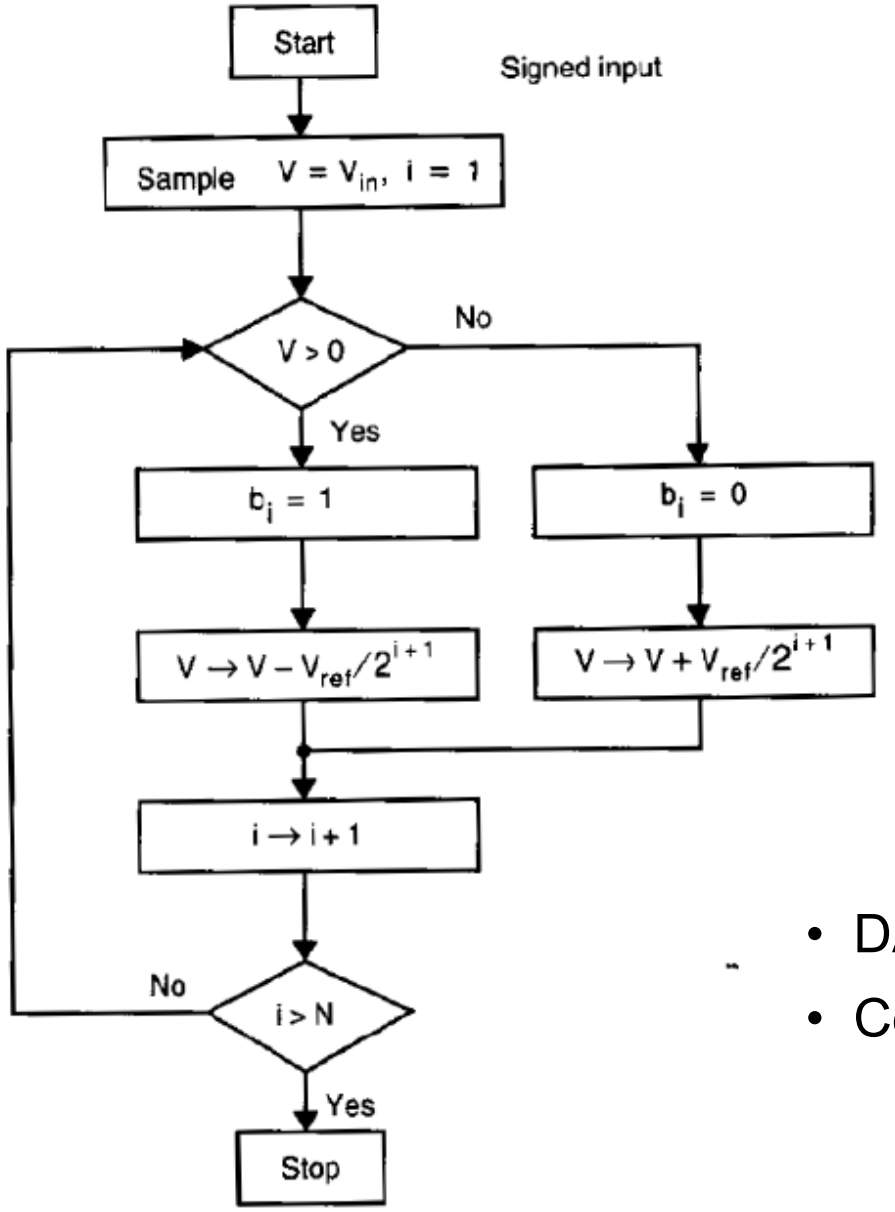
- Analogous to the algorithmic converter, but the iterations are all done in parallel in an assembly-line fashion.
- The latency is the same as the algorithmic A/D, but the throughput increases to the bandwidth of a single stage.

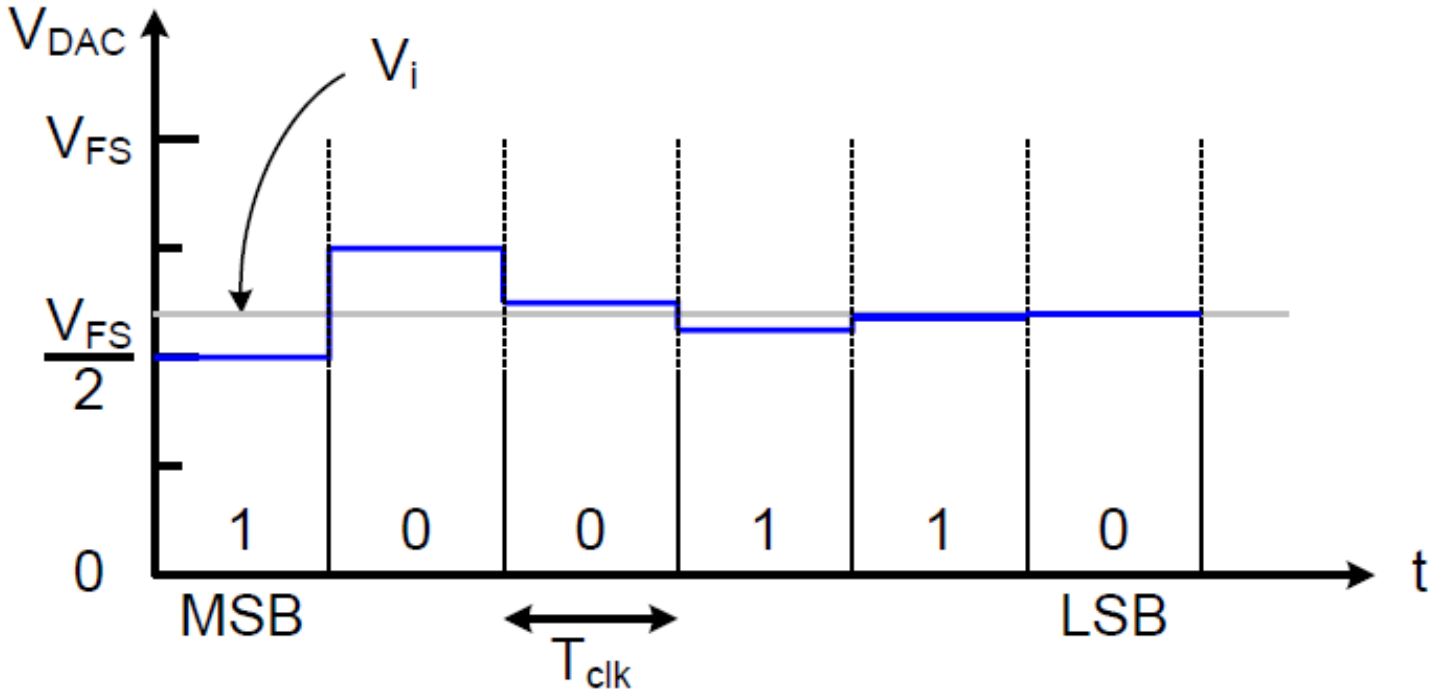
Properties of successive approximation converters



- Binary search over DAC inputs.
  - $N \times T_{clk}$ to complete N bits.
  - Successive approximation register or SAR.
- High accuracy achievable (16+ bits).
- Relies on highly accurate comparator.
- Moderate speeds (typically 0.1-100 MS/s).

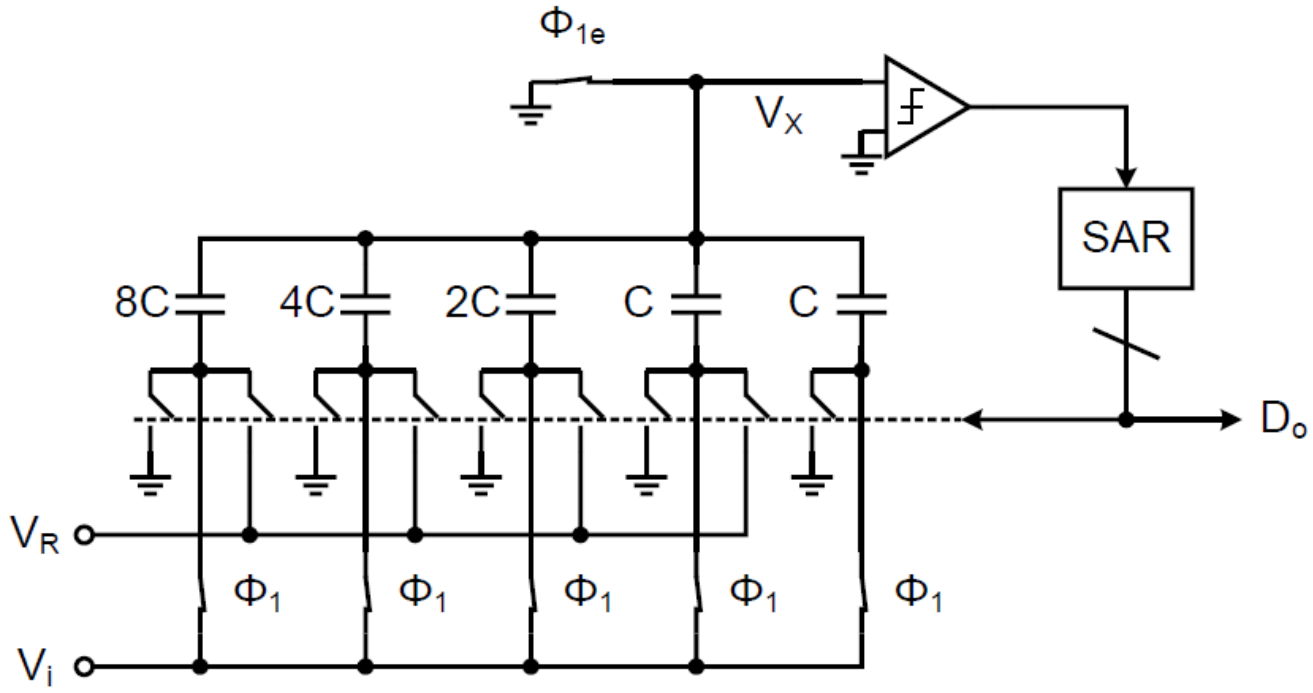
The binary search algorithm





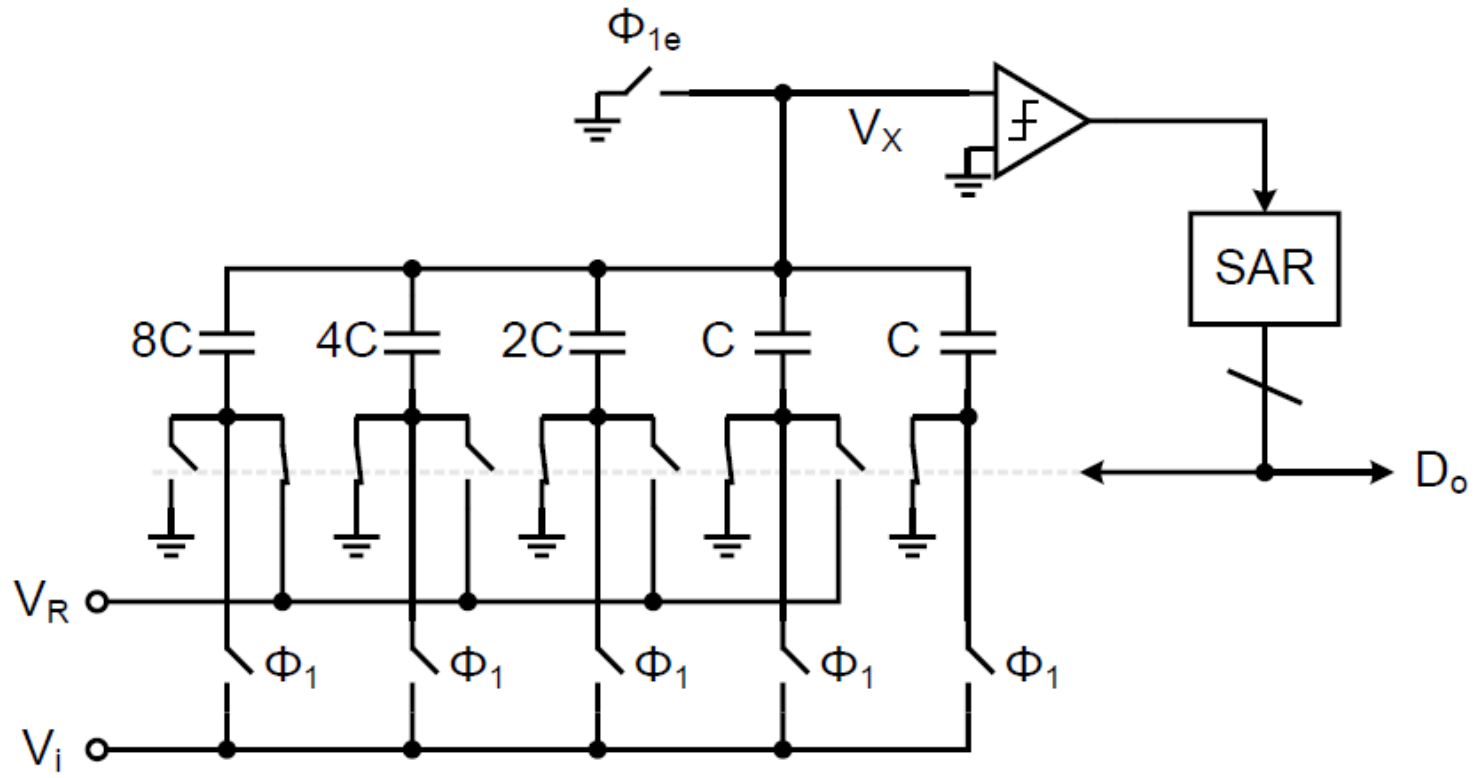
- DAC output gradually approaches the input voltage.
- Comparator differential input gradually approaches zero.

Charge redistribution SAR ADCs



- Most common SAR ADC architecture.
- Uses an *N*-bit binary-weighted capacitor array DAC (*aka* charge scaling DAC).
- Capacitor array samples input when Φ_1 is asserted (bottom-plate sampling).
- Comparator acts as a zero-crossing detector.
- Practical implementations are fully-differential.

4-bit example: Charge redistribution (MSB)

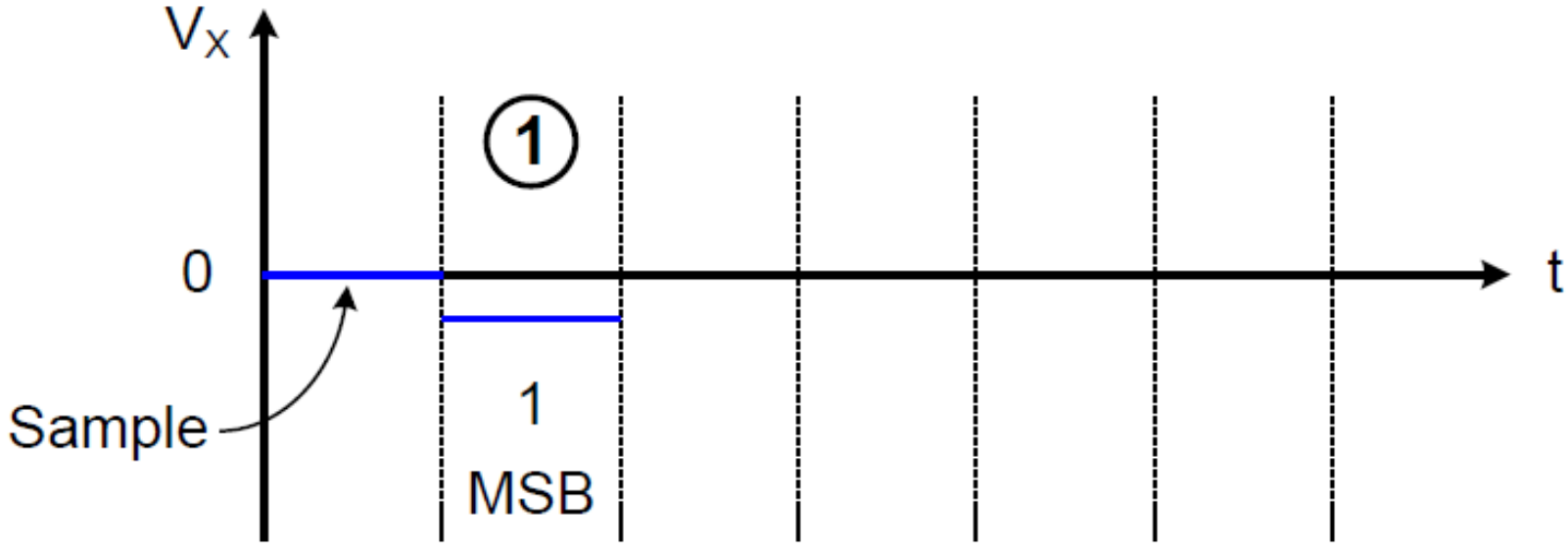


- Start with C_4 connected to V_R and the others to 0 (i.e., SAR = 1000).
- Now use charge conservation at V_X to find its new value:

$$V_i \cdot 16C = (V_R - V_x)C_4 - V_X(C_3 + C_2 + C_1 + C_0) \Rightarrow V_X = \frac{V_R}{2} - V_i.$$

Brookhaven National Laboratory

4-bit example: Comparison (MSB)

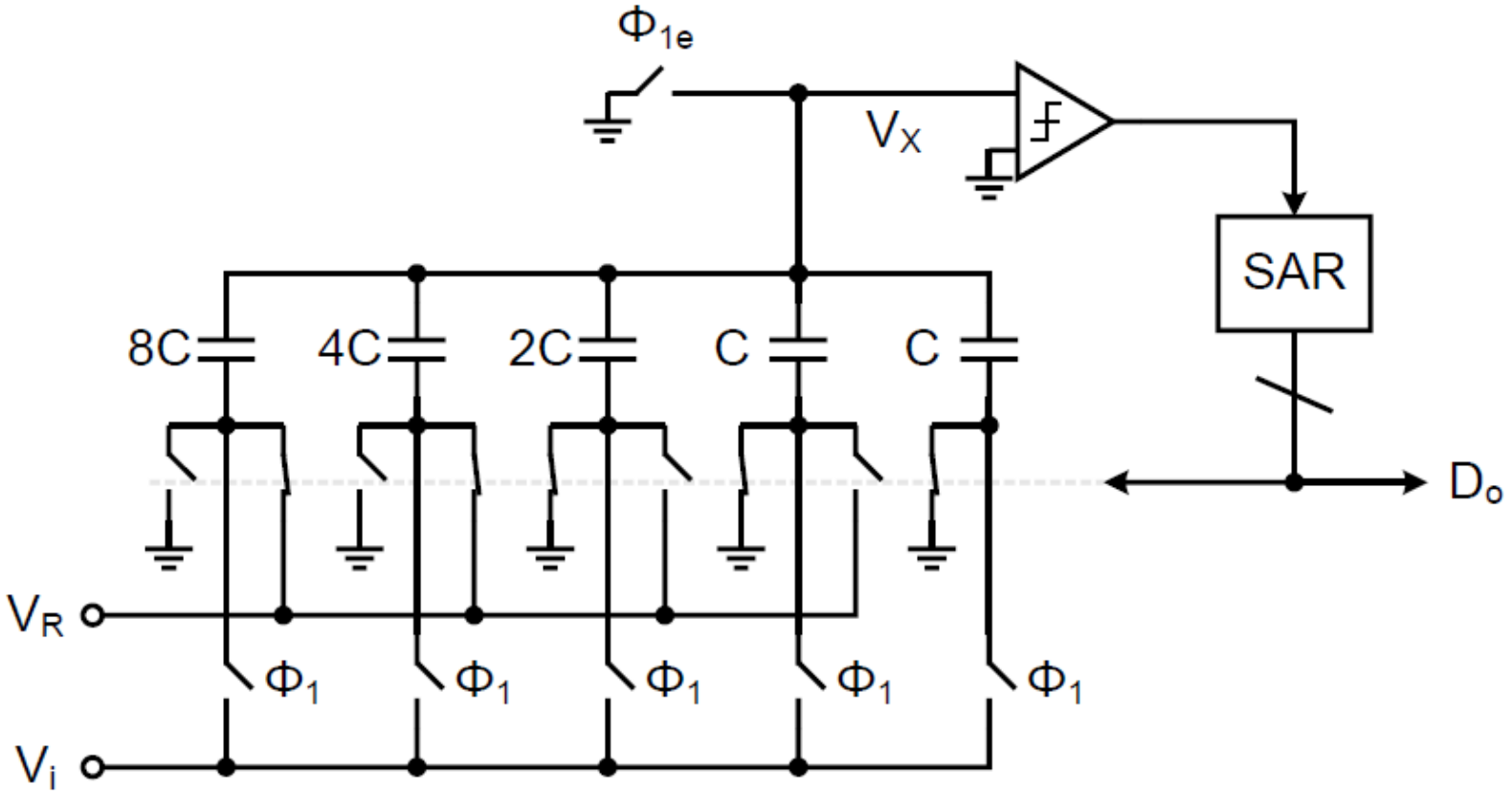


$$\text{MSB TEST: } V_X = \frac{V_R}{2} - V_i.$$

If $V_X < 0$, then $V_i > V_R/2$ and MSB = 1, so C_4 remains connected to V_R.

If $V_X > 0$, then $V_i < V_R/2$ and MSB = 0, so C_4 is switched to ground.

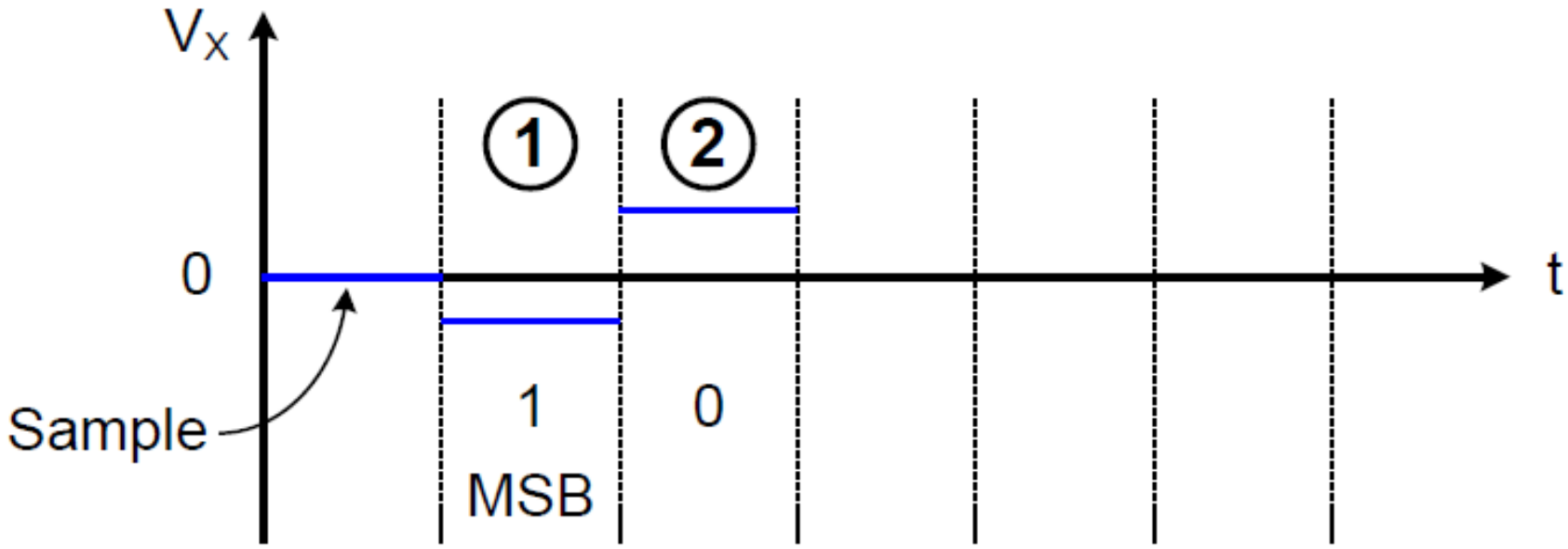
4-bit example: Charge redistribution (MSB-1)



- Assume SAR = 1100:

$$V_i \cdot 16C = \left(V_R - V_x\right) \cdot 12C - V_X \cdot 4C \Rightarrow V_X = \frac{V_R \cdot 12C - V_i \cdot 16C}{16C} = \frac{3}{4}V_R - V_i.$$

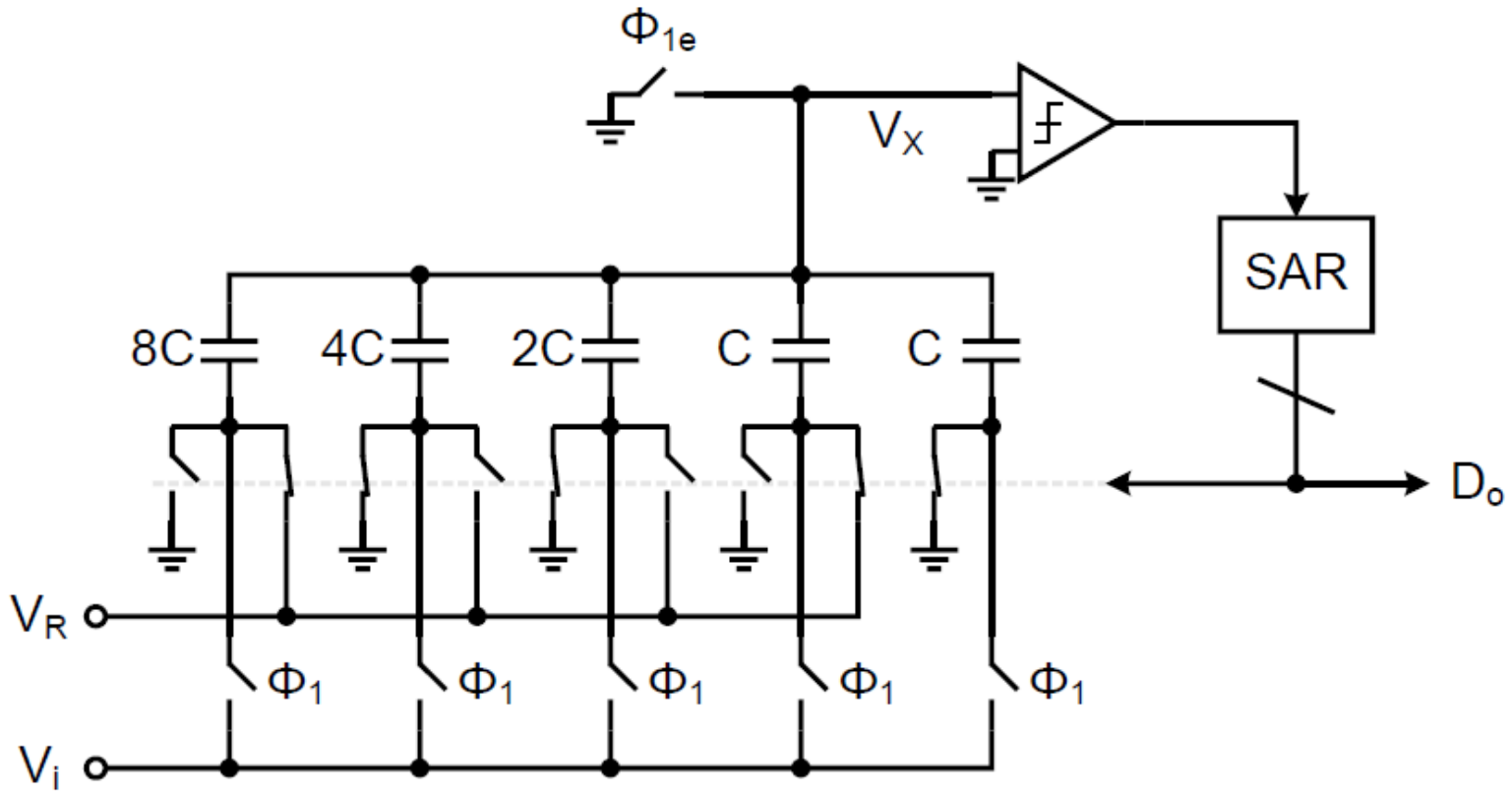
4-bit example: Comparison (MSB-1)



$$(\text{MSB}-1)\ \text{TEST: } V_X = \frac{3}{4}V_R - V_i.$$

- If $V_X < 0$, then $V_i > 3V_R/4$ and MSB−1 = 1, so C_3 remains connected to V_R.
- If $V_X > 0$, then $V_i < 3V_R/4$ and MSB−1 = 0, so C_3 is switched to ground.

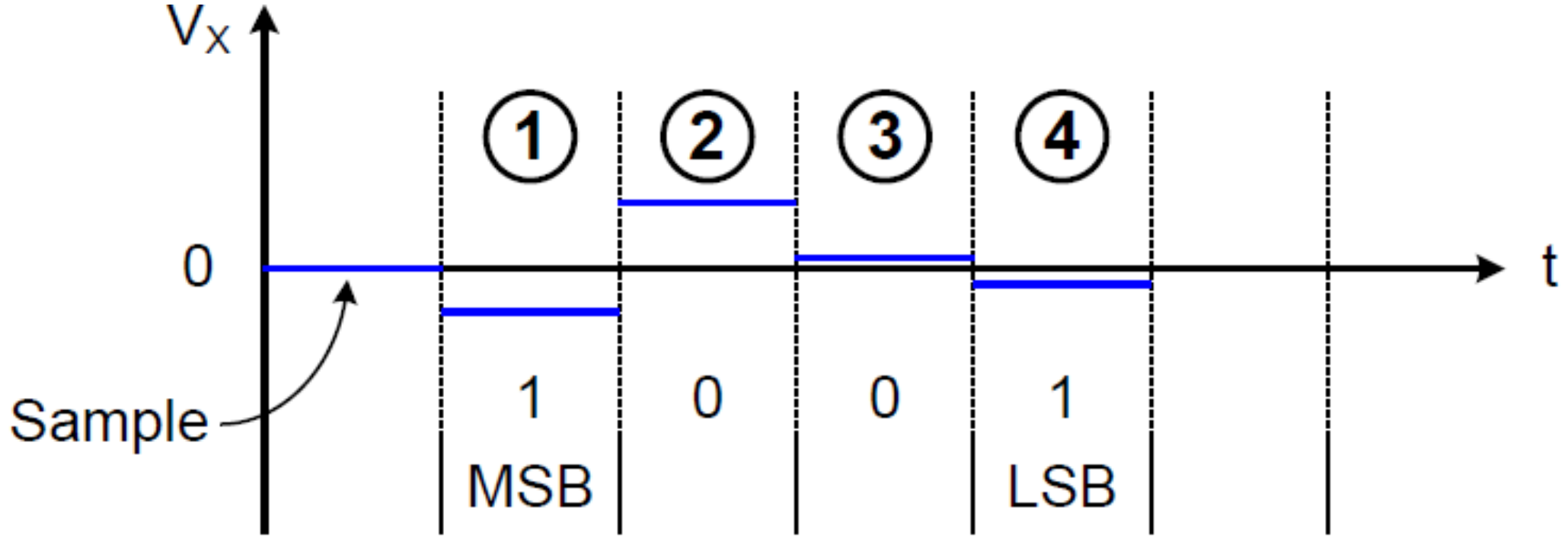
4-bit example: Charge redistribution (other bits)



- Assume SAR = 1010, and so on…
- The test completes when all four bits are determined using four charge redistributions and comparisons.

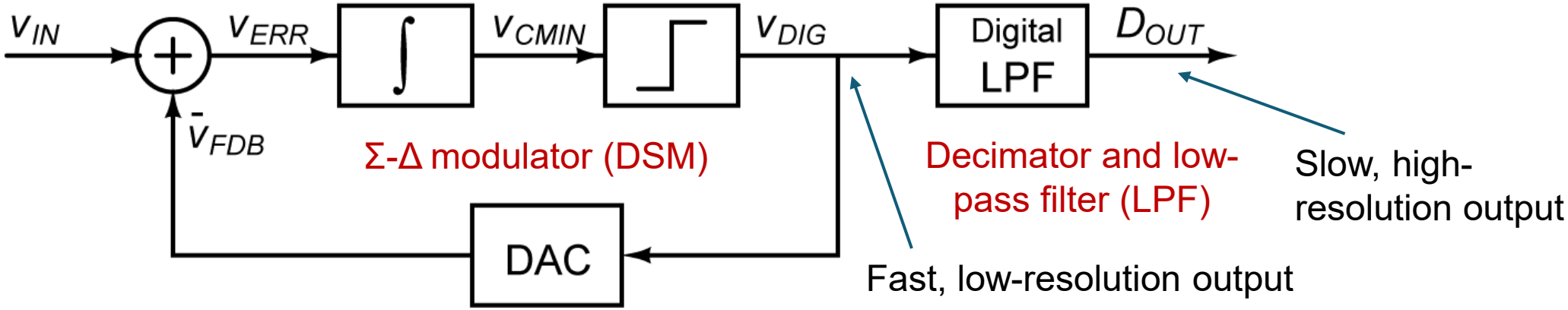
Brookhaven National Laboratory

After four clock cycles…



- Usually, about half of T_{clk} is allocated for charge redistribution and the other half for comparison and digital logic.
- V_X always converges to 0 (or V_{os} if the comparator has nonzero offset).

Sigma-delta (Σ-Δ) converters



- Sigma-delta converters sample at a rate f_s much faster than the Nyquist rate (which is set to f_0 by the output LPF) and use **averaging and noise shaping to reduce quantization noise and achieve higher precision.**
- The effective quantization noise σ_q of a 1-bit quantizer is decreased by a factor of

$$(f_s/2f_0)^{3/2} = (f_s/2f_0)^{1/2} \times (f_s/2f_0).$$

- In general, if we have N integrators in the loop filter, σ_q is reduced by a factor of

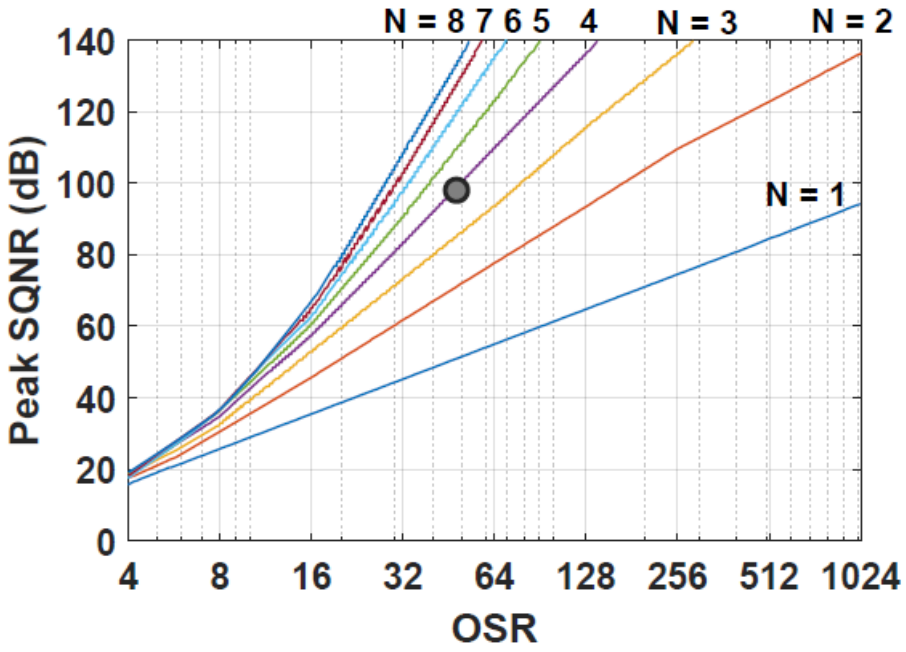
$$(f_s/2f_0)^{N+1/2} = (f_s/2f_0)^{1/2} \times (f_s/2f_0)^{N}.$$

- Here $f_s/2f_0$ is known as the oversampling ratio (OSR).

Why is the quantization noise reduced?

$$\sigma_q \propto (OSR)^{-(N+\frac{1}{2})} = (OSR)^{-1/2} \times (OSR)^{-N}$$

- The square-root factor in quantization noise arises from averaging: The quantization noise ***power*** per unit bandwidth is given by $\Delta^2/(12f_s)$, so the total noise power is given by $\Delta^2 f_0/(12f_s)$.
- Another factor of $(OSR)^N$ reduction comes about because of noise shaping: the quantization noise of the comparator at the output of the feedback loop is reduced by $1/(1+A(s))$, where $A(s)$ is the loop transmission. Since $A(s)$ scales as the integrator transfer function $1/(\tau s)^N$, the quantization noise is differentiated; it acquires a high-pass shape of $(\tau s)^N/((\tau s)^N+1)$ that is removed by the digital LPF.
- Quantization noise can be reduced even further with multi-bit quantizers. Σ-Δ converters scale well with technology and have become quite energy efficient.



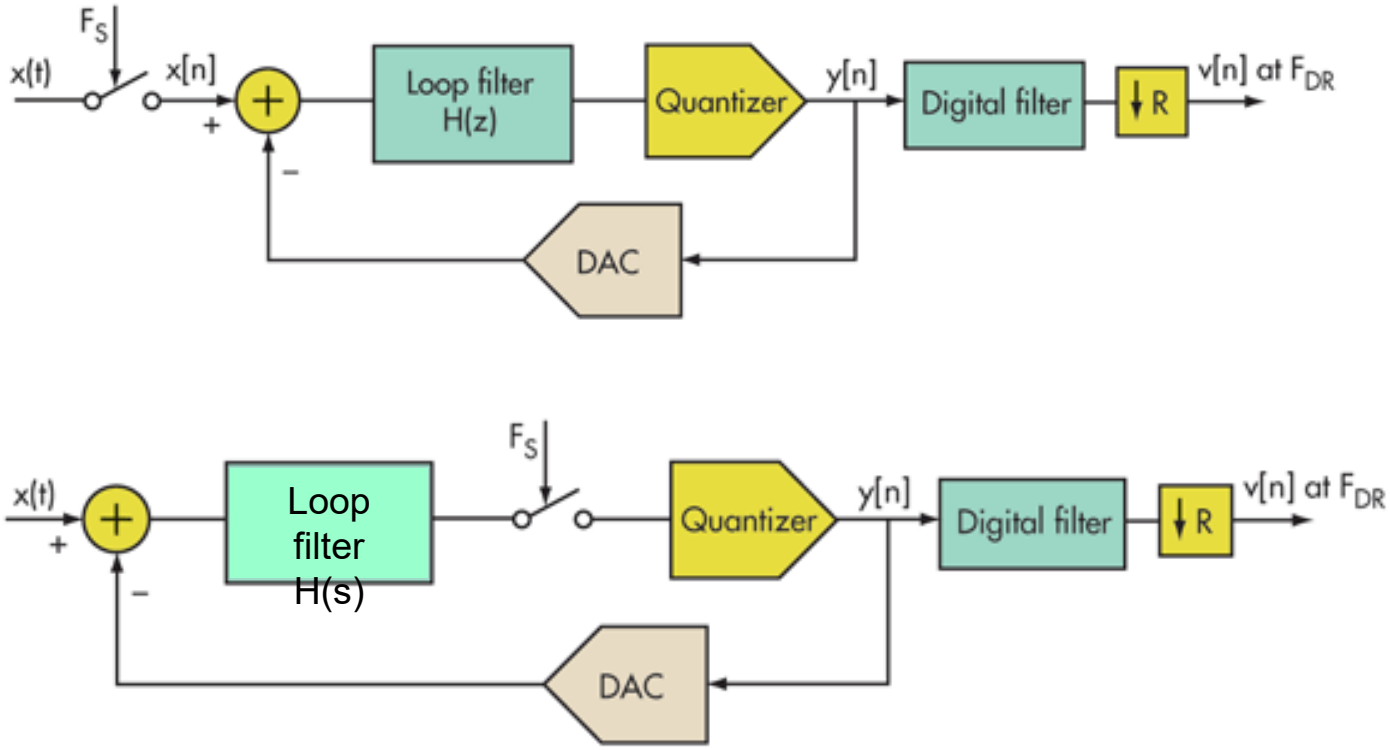
Available peak SQNR for DSMs of various orders (N = 1 to 8) as a function of OSR. The grey dot represents an example design (N = 4, OSR = 50) with peak SQNR of 95 dB (about 16 bits).

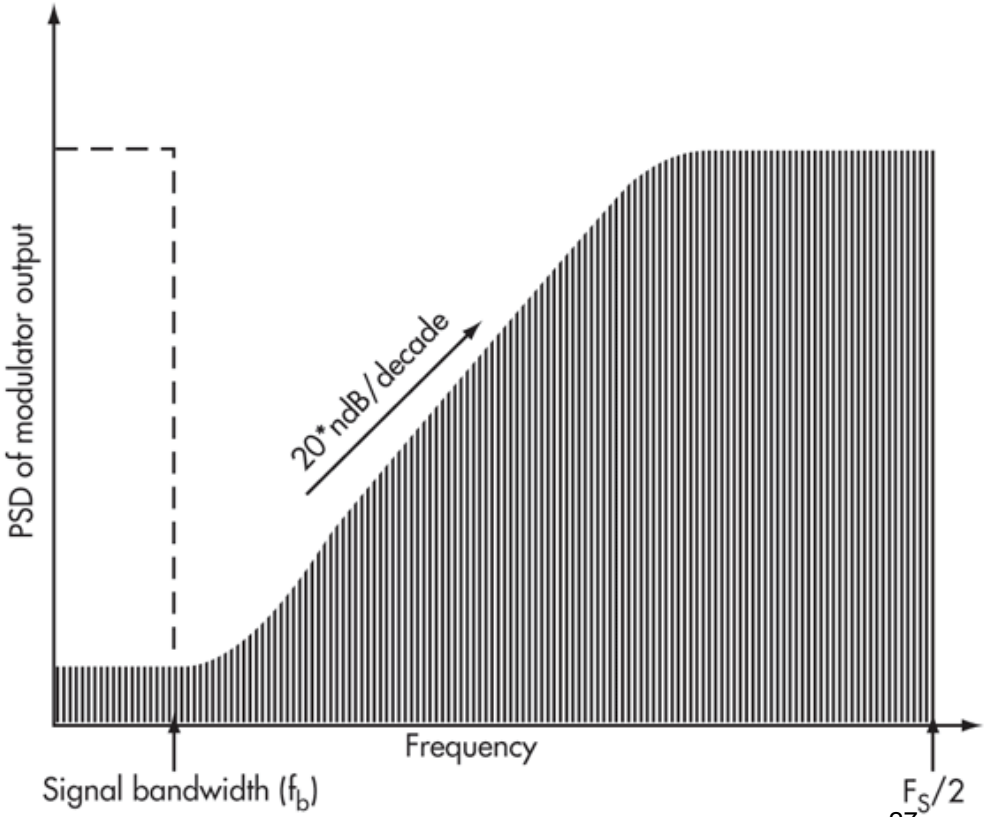
SQNR = Signal to quantization noise ratio

Brookhaven National Laboratory

Sigma-delta (Σ-Δ) converters (2)

- In a discrete-time Σ-Δ, sampling occurs before the loop filter. This is the most popular topology for signal bandwidths up to about 5 MHz.
- In a continuous-time Σ-Δ, sampling occurs just before the quantizer. This topology is useful for sampling wideband signals.





Brookhaven National Laboratory

Time-to-digital converters (TDCs)

- Used to quantize the time between two events.
- Traditional approach: Convert time to voltage, then voltage to time (via an ADC).
- More modern approach: Quantize time directly. The simplest design is analogous to a flash ADC.

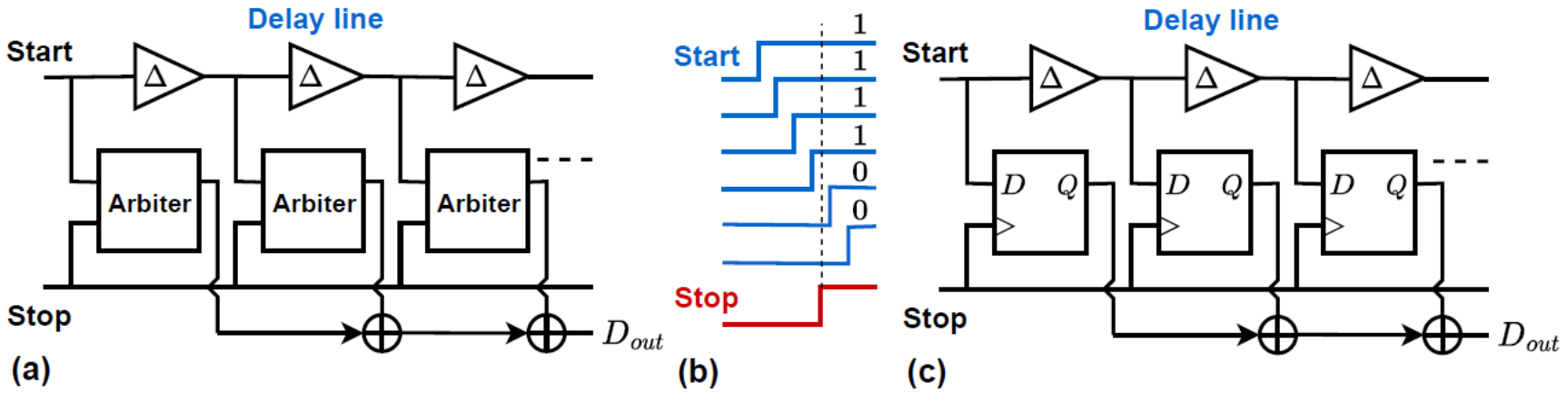


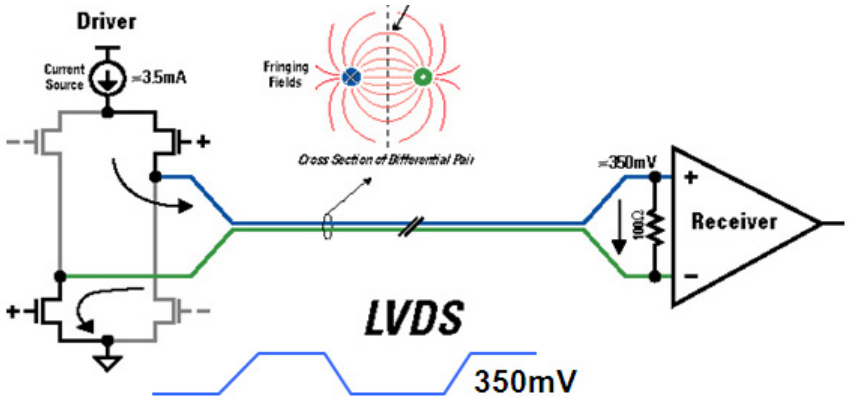
Figure 17.51: (a) Generic schematic of a flash (or delay line) TDC. The digital adders convert the thermometer-coded outputs of the arbiters to binary form. (b) Major waveforms within the flash TDC shown in (a): the arbiters sample the delayed start signals when the stop signal goes high. (c) Practical implementation of the flash TDC using D-type flip-flops as arbiters.

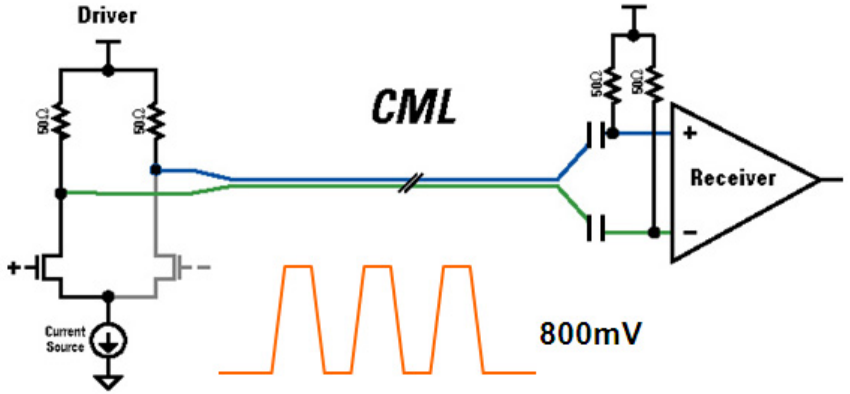
Digital data transmission

Soumyajit Mandal, Nicholas St. John

Motivation

- Digitized outputs must be transmitted off-chip, often on long transmission lines.
  - The associated circuit is known as a wireline transmitter or line driver.
  - Transmitted signals are generally differential to reject common-mode pickup.
  - Many protocols: low-voltage differential signaling (LVDS), current-mode logic (CML)…

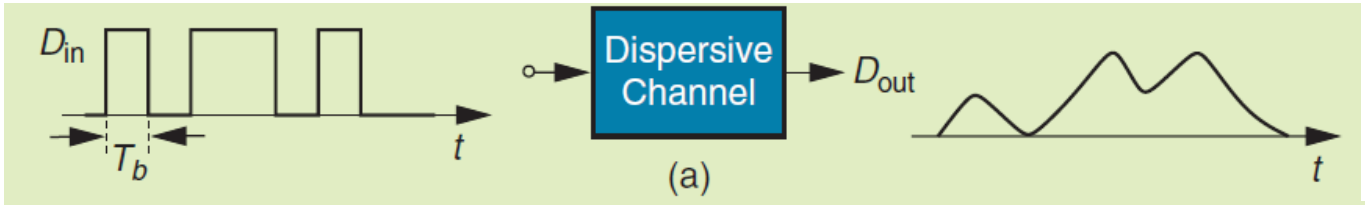


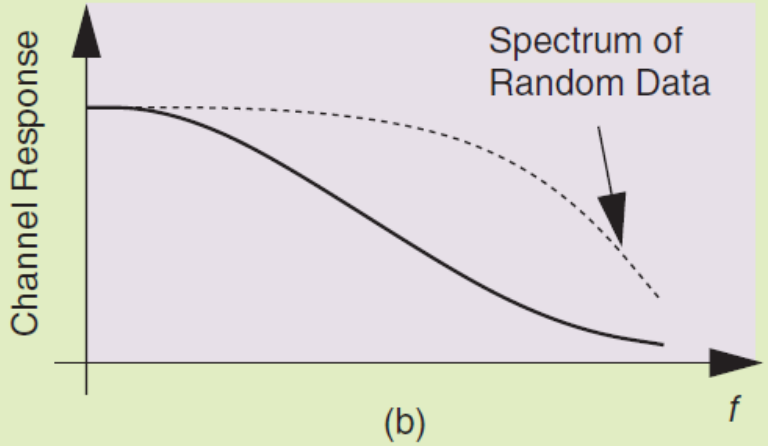


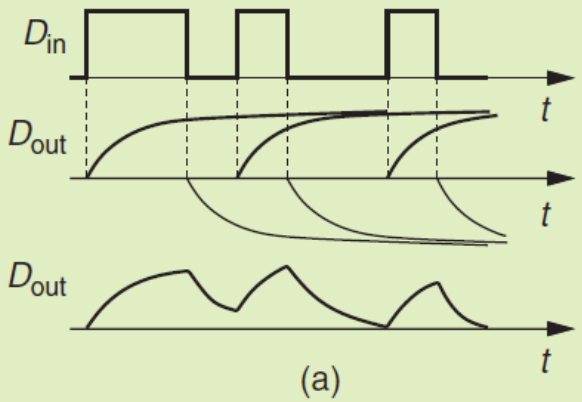
- Some experiments require line drivers capable of transmitting across high loss radio-pure cables.
  - Pre-emphasis is a technique for adequately driving such cables.
  - Line drivers with pre-emphasis are typically designed or a given application and cable type.
  - Rather than designing a custom line driver for each ASIC, a single block with configurable parameters can be used to provide good results with a variety of cables.

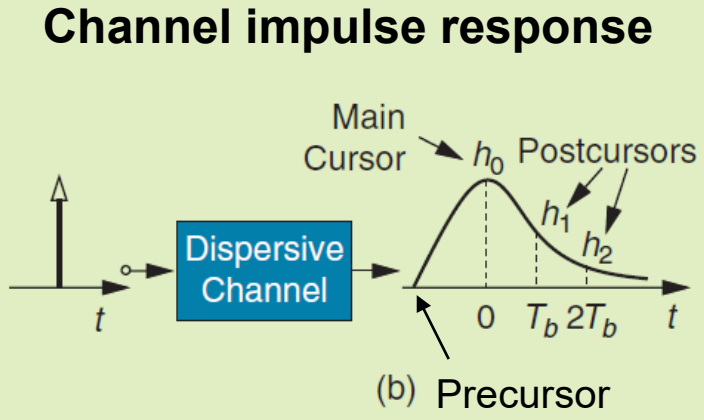
Effects of frequency-dependent channels on digital data

- Limited channel bandwidth results in attenuation, data-dependent jitter, and inter-symbol interference (ISI).







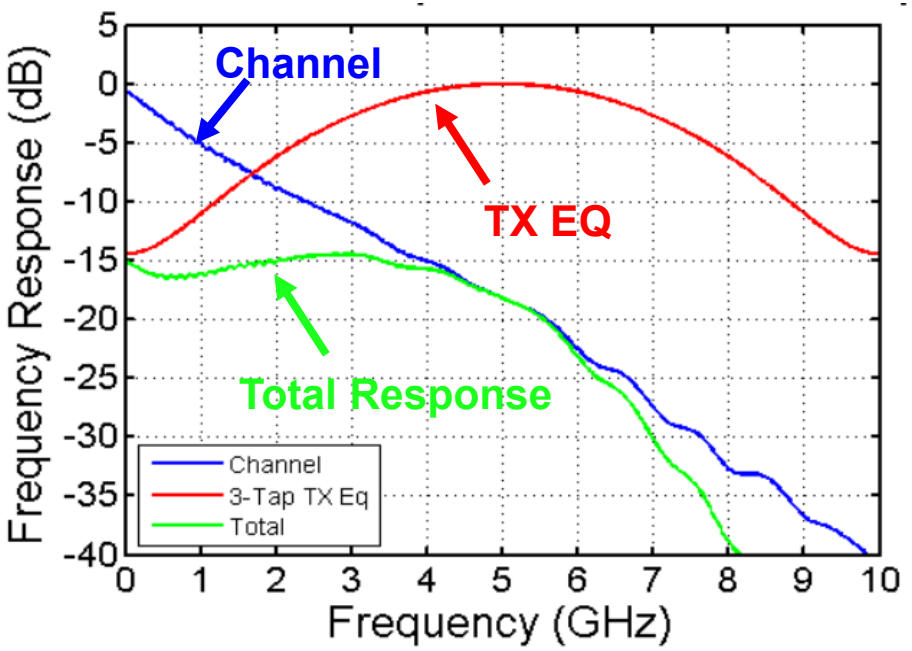


- Since the channel is linear, it can be modeled either in the time-domain (using the impulse response $h(t)$) or in the frequency-domain (using the transfer function $H(f)$).

Behzad Razavi, *IEEE Solid-State Circuits Magazine*, 2017

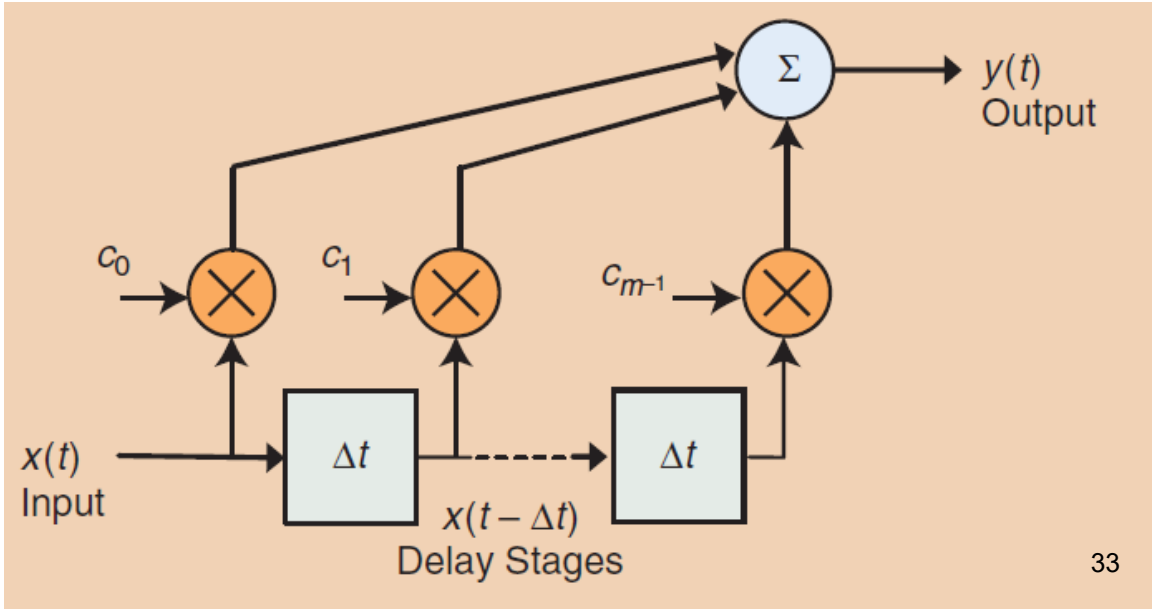
What is pre-emphasis?

- Pre-emphasis, also known as feed-forward equalization (FFE) is a method in which the waveform of transmitted data is manipulated to cancel out the filtering effects of cable channels.
- Cables can have very complex transfer functions $H(f)$, but these are generally low-pass in nature. Thus, pre-emphasis aims to add high frequency components to the data.



Implementing pre-emphasis

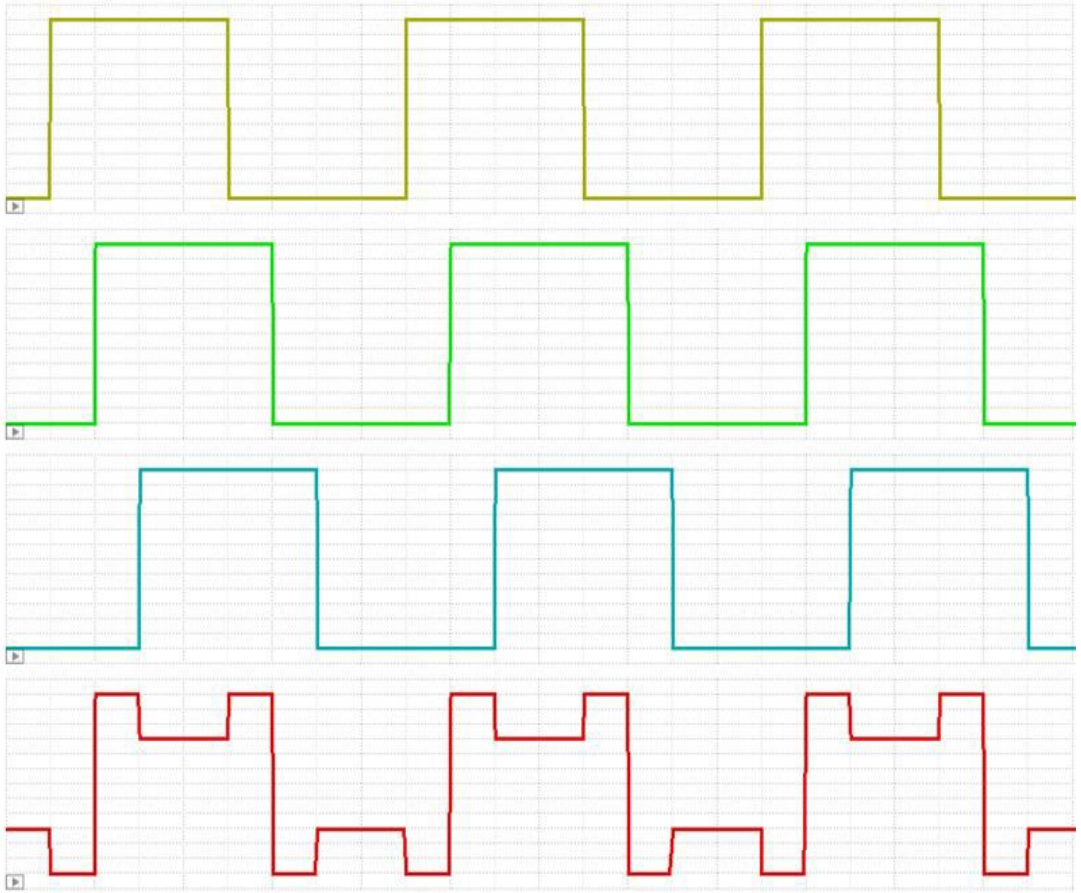
- Pre-Emphasis is generally implemented via a Finite Impulse Response (FIR) filter as shown.
- In the diagram, *m* taps are implemented with relative delays of Δt.
- Each tap has its own independent weighting factor, c_i, and all taps are then summed together.
- Typically, each c_i is configurable, but Δt is usually hard-coded to one unit interval (UI)*.
- More generally, both c_i and the time delays Δt can be individually programmable.



*1 UI = 1 symbol period

John F. Bulzacchelli, “Equalization for electrical links: current design techniques and future directions.” *IEEE SSCS Magazine* 7.4 (2015).

Pre-emphasis in the time domain

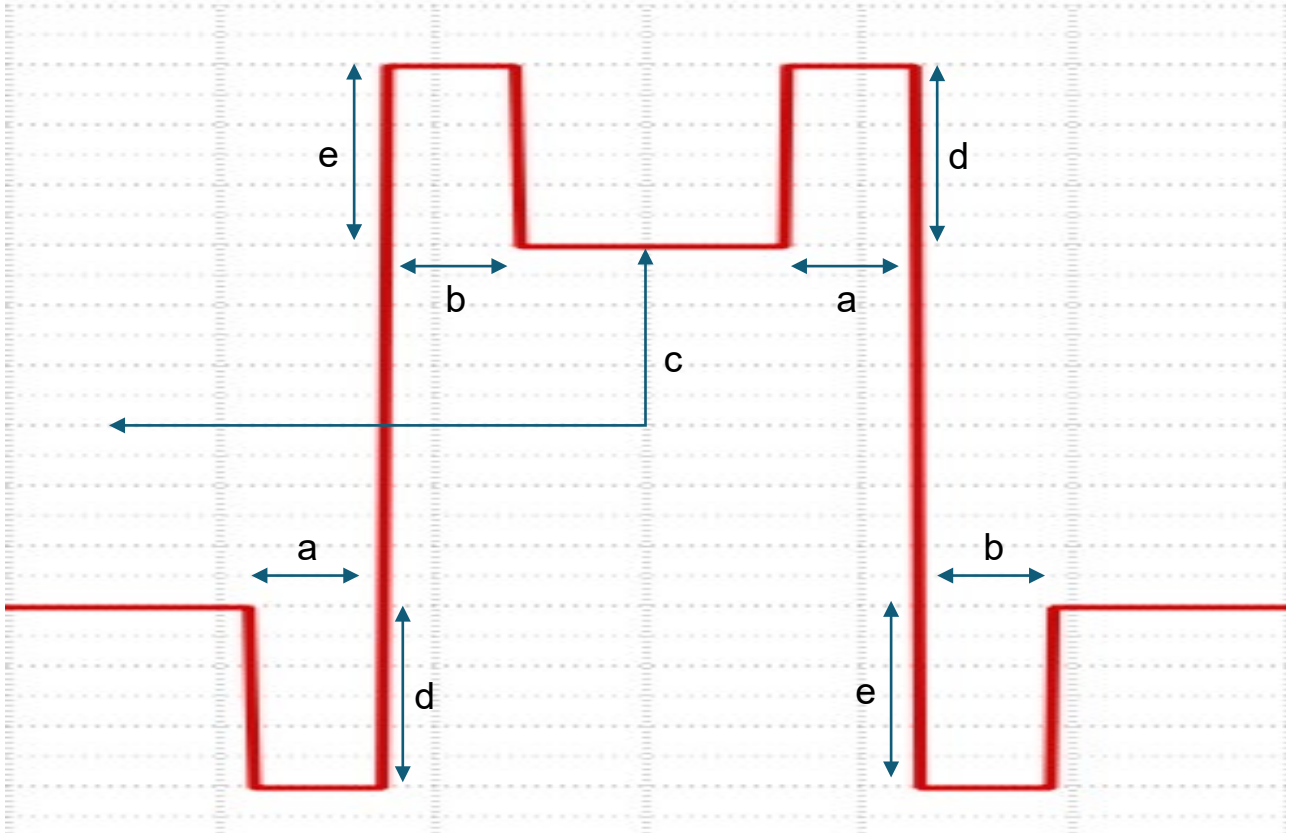


In the figure, a 3-tap FIR filter implements generalized pre-emphasis, for which:

- Yellow waveform is the Pre-Data tap
- Green waveform is the Main-Data tap
- Blue waveform is the Post-Data tap
- Red waveform is the output waveform, which is the weighted sum of the three taps.

Pre-emphasis in the time domain (2)

- The figure shows the generic output waveform shape for a line driver using generalized pre-emphasis.
- Ideally, each of the parameters, *a, b, c, d* and *e*, should be configurable.



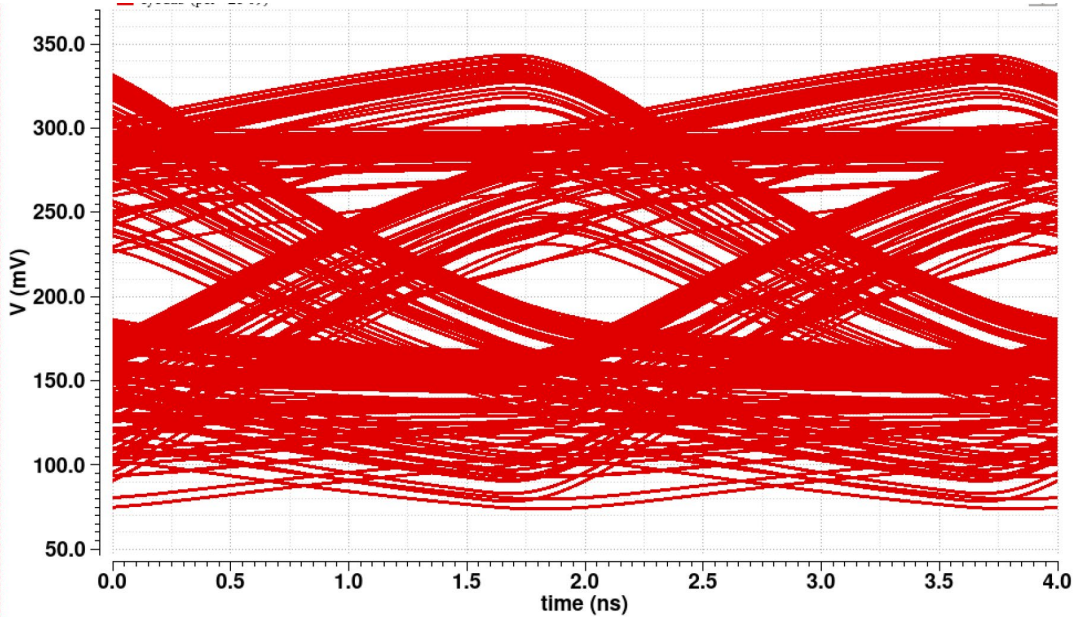
- a: Duration of pre-emphasis prior to edge transition
- b: Duration of pre-emphasis after an edge transition
- c: Normal logic level (no pre-emphasis)
- d: Pre-emphasized level prior to edge transition
- e: Pre-emphasized level after edge transition

Effects of configuring pre-emphasis settings

- Performance studied by plotting eye diagrams (synchronized overlay of all possible data transitions).
- The bit-error rate (BER) improves exponentially as the height and width of the eye opening increases.

Normal settings, high-loss cable:

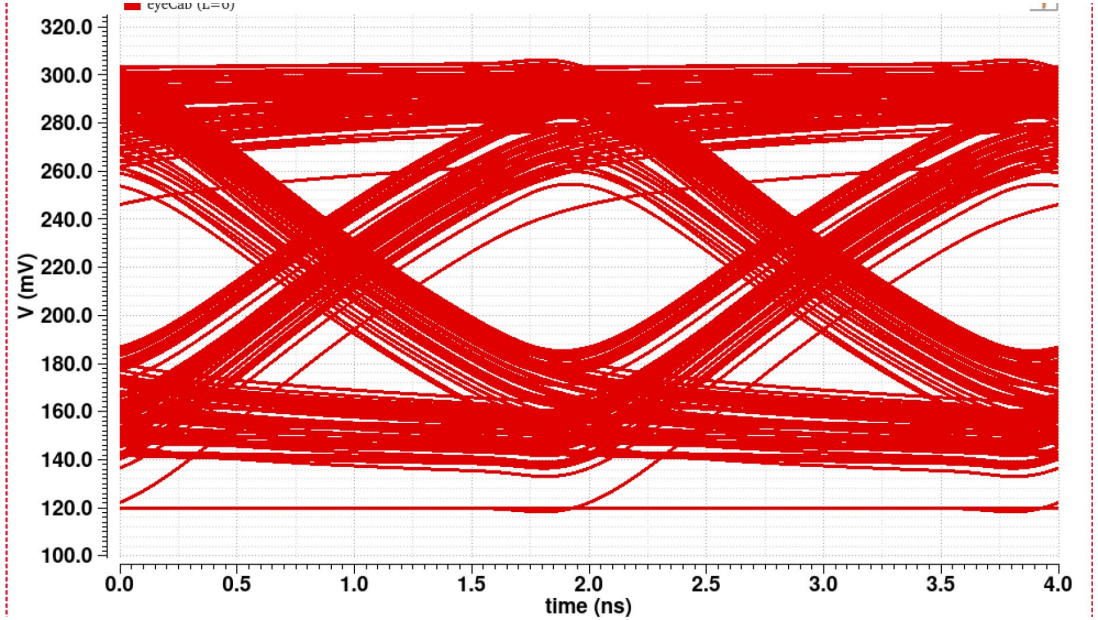
Pre-Tap Duration: 100% of UI | Post-Tap Duration: 100% of UI



Eye Opening: 37.6mV and 0.879ns

Optimized settings, same cable:

Pre-Tap Duration: 10% of UI | Post-Tap Duration: 90% of UI



Eye Opening: 74.4mV and 1.183ns

Input: 7-bit pseudo-random bit sequence (PRBS), bit period of 2ns (500 Mb/s)

Tap Weights: 600mV each

Effects of impedance matching

- In addition to slowly-varying frequency-dependent loss, the channel transfer function (TF) can include narrowband features (peaks/nulls) due to reflections from impedance discontinuities.
  - These narrowband features can significantly degrade ISI and thus BER.
  - A common source of reflections is mismatch between the channel (modeled as a transmission line with impedance Z_0) and the transmitter/receiver circuit (source/load impedances of Z_S and Z_L, respectively).
  - The source and load reflection coefficients are defined as:

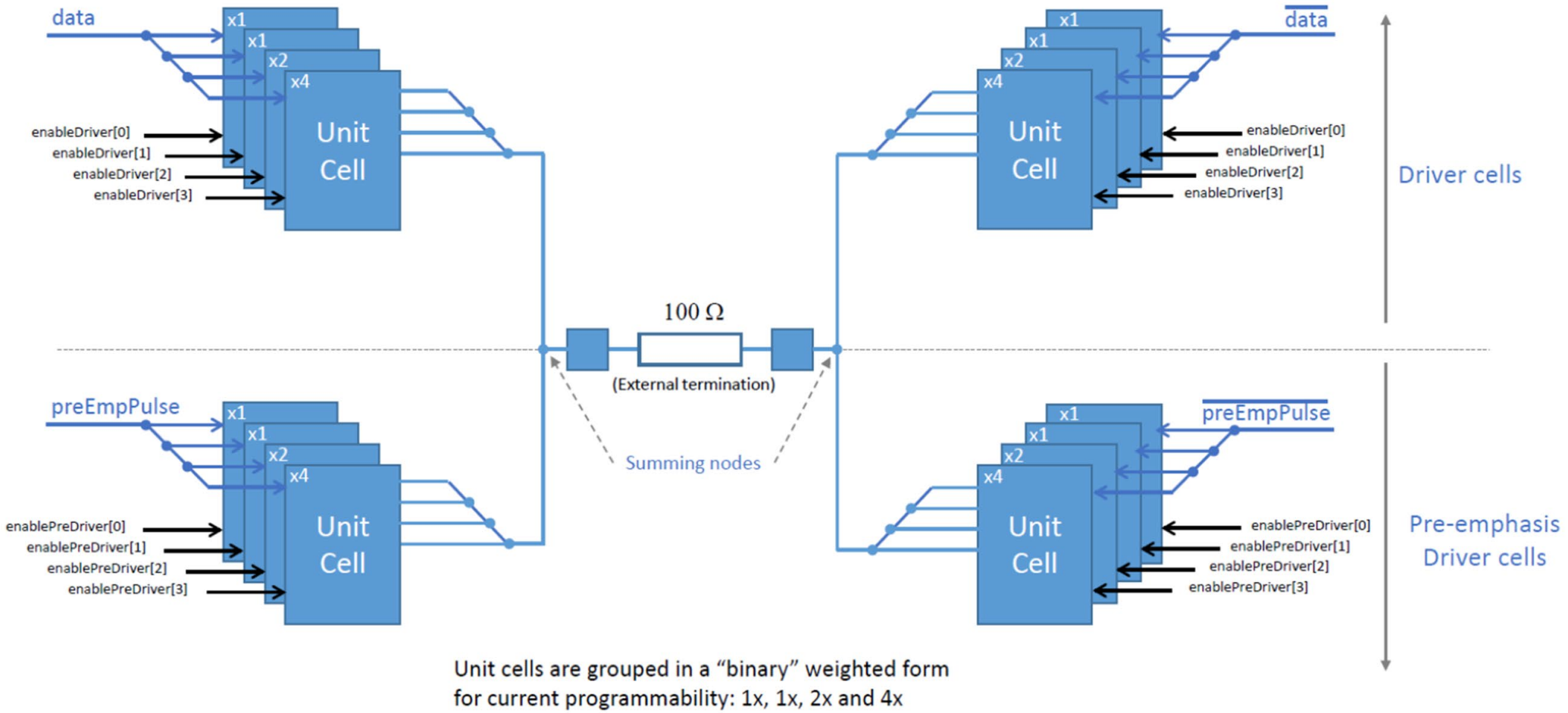
$$\Gamma_S = \frac{Z_S - Z_0}{Z_S + Z_0}, \quad \Gamma_L = \frac{Z_L - Z_0}{Z_L + Z_0}$$

Amplitude of reflected signal after one round trip $\propto \Gamma_S \Gamma_L H^2(f)$.

- Reflected signals can be eliminated by impedance matching *either* the source ($\Gamma_S = 0$) or the load ($\Gamma_L = 0$).
- For a high-loss channel ($|H(f)| << 1$), impedance matching is not necessary to eliminate reflections.
  - In this case, it is better to use a low-impedance source ($|Z_S| << Z_0$) and high-impedance load ($|Z_L| >> Z_0$) to maximize the received voltage amplitude and signal-to-noise ratio (SNR).

Razavi, *IEEE Solid-State Circuits Magazine*, 2017

Example of a line driver design: the lpGBT

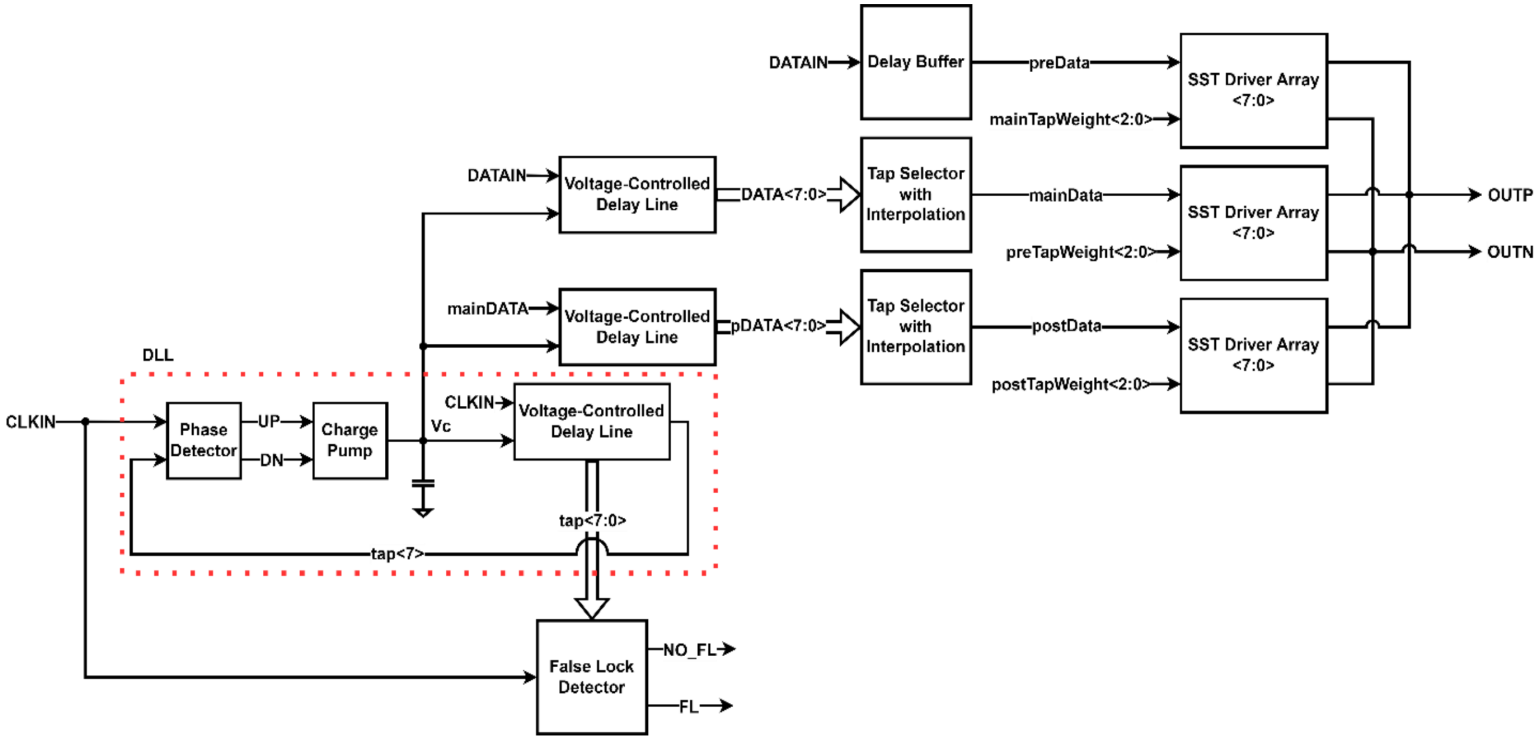


Unit cells are grouped in a "binary" weighted form for current programmability: 1x, 1x, 2x and 4x

Line driver with generalized pre-emphasis

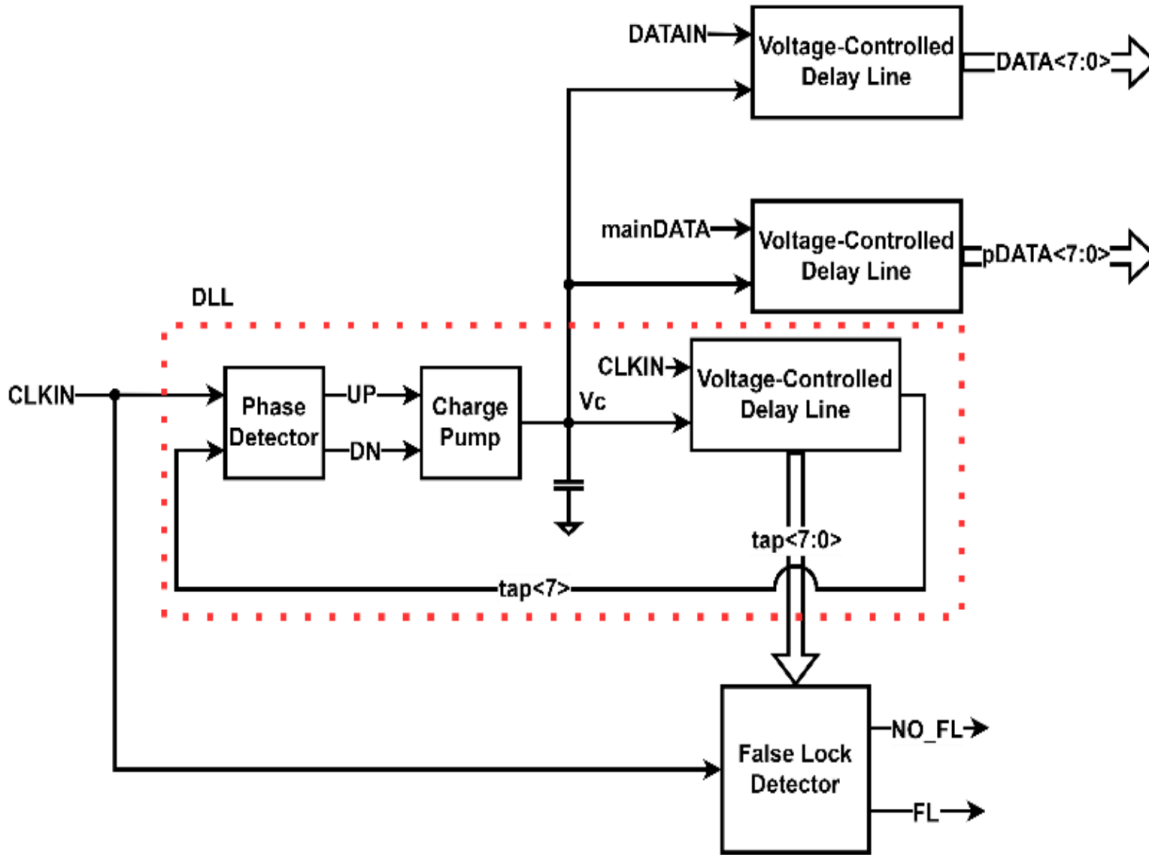
Main building blocks:

- An *N*-stage Delay-Locked Loop (DLL) with False Lock Detection.
- Replica Voltage-Controlled Delay Lines (VCDLs) that utilize the DLL to create delayed versions of the input data.
- Digital Interpolator blocks that allow the user to select a delayed data copy or interpolate two adjacent copies, giving the user a total of 2*N* delay options.
- Three programmable arrays of Source-Series Terminated (SST) drivers for power-efficient transmission and adjustable weights for each tap with acceptable impedance matching.



DLL and Replica VCDL Behavior

- The DLL compares the system clock with the final output of the VCDL using a Phase Detector (PD).
- The PD output then controls a Charge Pump (CP) via UP and DOWN logic signals.
- The CP output voltage, V_c, is a DC control for the VCDL, either increasing or decreasing the relative delay of all *N* delay cells in the line (*N* = 8 in this example).
- The system locks when tap<N> is aligned with CLKIN
  - This occurs when tap<N> is delayed from CLKIN by n*UI where n is a positive integer.
  - A False Lock Detector can be used to ensure that n = 1.
- V_c also controls two replica VCDLs outside the DLL that each create *N* delayed data copies for the user to select.



Detecting and compensating for false locking

- False locking occurs when the 2π periodicity of the PD output causes the final VCDL tap to be delayed from the input by n*UI (where n is greater than 1) in steady-state.
- Such a false lock point is not ideal, because an *N*-stage DLL normally creates *N* unique output taps, but false locking degrades the number of unique taps to *N*/n.
- Typical false lock detection circuits run continuously and utilize frequency multiplication
  - This adds a lot of switching power dissipation and is unnecessary as the system needs to only be checked once after locking is achieved.
- A single-test false lock detector is preferable. Such a circuit only checks for false-locking at the request of the user.
  - After testing, the detector can shut itself down to preserve power.

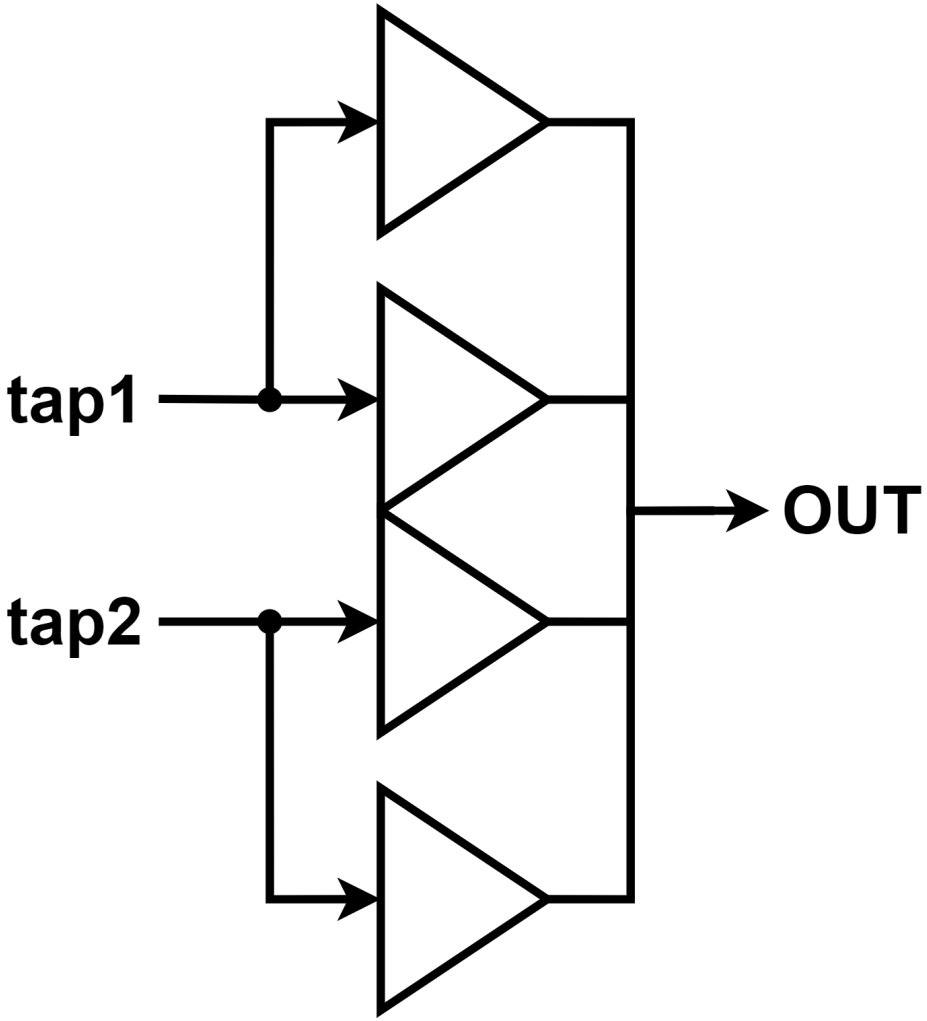
False lock detection test

- To understand what tests need to be run, assume *N* = 8 and consider the table below.
- When the tap delay is equal to a whole number, that tap is aligned with the input.
- For each false locking scenario (n != 1) we see that we can test tap<3:0> and tap<7>
  - Tap<7> must be aligned with the input.
  - The relative delays (modulo 1 x UI) of the taps from the input must be in this order from least to greatest: tap<0>, tap<1>, tap<2> and tap<3>.
- If these two aspects are met, then n = 1
  - Else, there is a false lock

	Delay of Tap to Input Divided by UI							
n	tap<0>	tap<1>	tap<2>	tap<3>	tap<4>	tap<5>	tap<6>	tap<7>
0.5	0.0625	0.125	0.1875	0.25	0.3125	0.375	0.4375	0.5
1	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
2	0.25	0.5	0.75	1	1.25	1.5	1.75	2
3	0.375	0.75	1.125	1.5	1.875	2.25	2.625	3
4	0.5	1	1.5	2	2.5	3	3.5	4
5	0.625	1.25	1.875	2.5	3.125	3.75	4.375	5
6	0.75	1.5	2.25	3	3.75	4.5	5.25	6
7	0.875	1.75	2.625	3.5	4.375	5.25	6.125	7

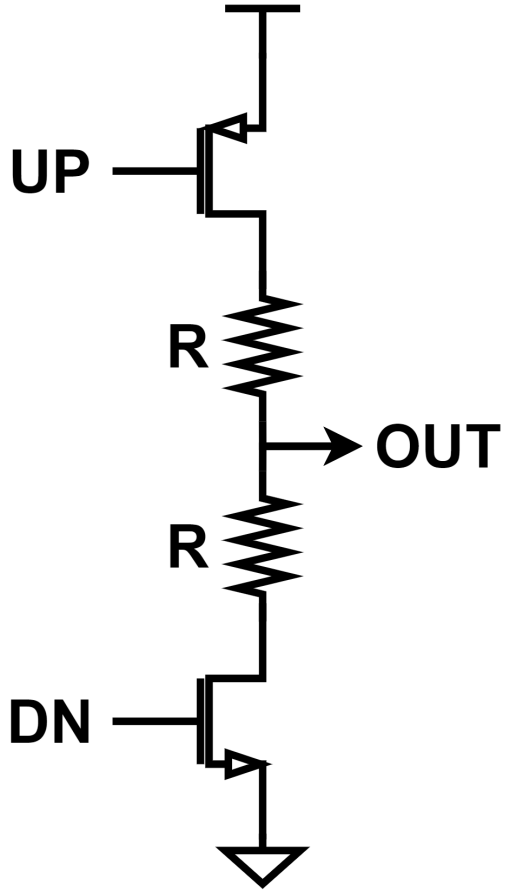
Tap selection with digital interpolation

- A digital interpolator can be used to allow the user to select the output DLL tap.
- Rather than adding more output taps to the DLL, which will make the DLL more difficult to run at higher data rates, interpolation increases the number of possible delays for a given number of taps.
- A single-stage interpolator can create delays halfway between its inputs, doubling the number of possible delays from *N* to 2*N*.
  - Multiple interpolation stages can be cascaded to further increase the number of delays.



Source-series terminated (SST) driver

- Driver circuits are usually differential and either behave as a high-impedance current source or a low-impedance voltage source.
- Current-mode drivers are preferable for impedance matching but consume more power for the same output signal swing.
- The common-mode level and signal swing of current-mode drivers are generally designed to satisfy industry standards such as current-mode logic (CML) or low-voltage differential signaling (LVDS).
- Voltage-mode drivers often use the SST design shown here.
  - Ideally, voltage-mode operation reduces power by 2-4x compared to current-mode.
  - However, voltage-mode drivers load one another when connected in parallel, thus affecting output impedance matching.
  - The value of *R*, the series resistance, can be made selectable to allow user adjustment of the output impedance and signal swing.



An SST driver “slice” (single-ended)

Example: Wired data transmission for nEXO

- High-loss, flexible, radio-pure cable (6-9 m) operating within a liquid Xenon TPC (at 165K).
- Required data rate: 250-500 Mb/s
- Need to minimize power consumption to minimize cryogen boil-off

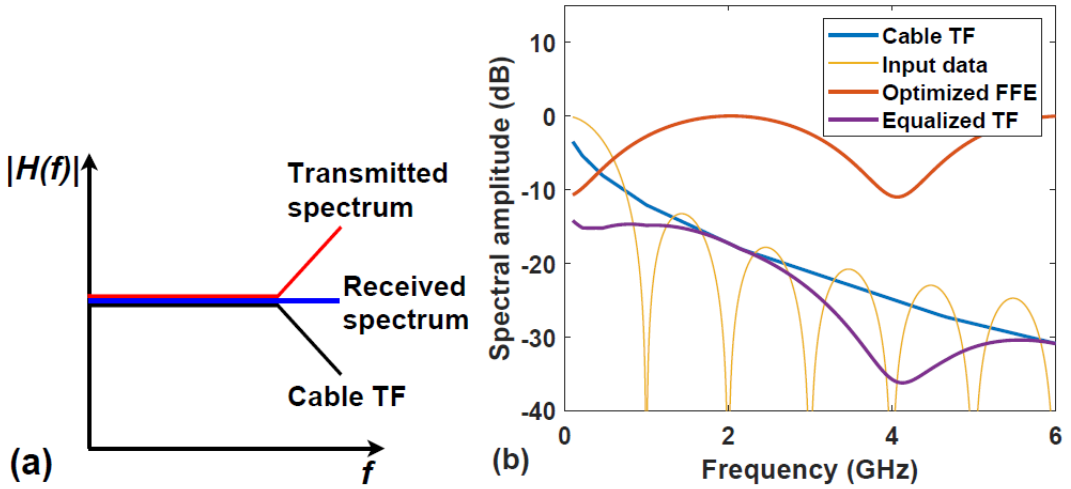


Fig. 2. (a) In an ideal case, pre-emphasis inverts the channel TF such that the spectrum of received data becomes flat within the transmit bandwidth. (b) A more realistic example, showing the use of a $K = 3$ tap FFE to invert the TF of a 2 m long radio-pure stripline fabricated on a flexible PCB.

N. St. John et al., *IEEE PRIME* (2023)

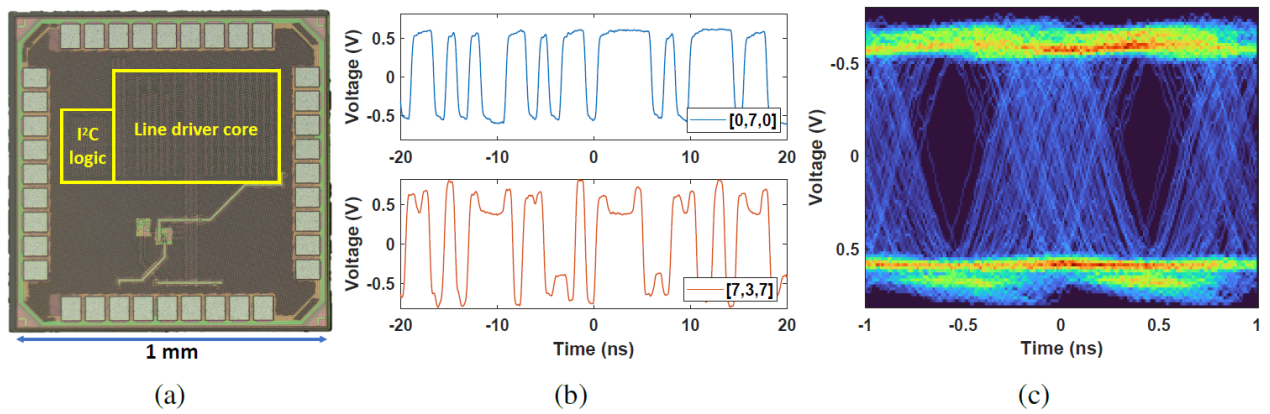


Figure 4: (a) Labeled die photograph of the test chip (total area = 1 mm^2). (b) Measured driver waveforms (PRBS-31 data) at 0.8 Gb/s after 0.5 m of coaxial cable for two different pre-emphasis settings (quantified by the weight vector $w = [w_{pre}, w_{main}, w_{post}]$): $w_{pre} = w_{post} = 0$, resulting in no pre-emphasis (top); and $(w_{pre} = w_{post}) > w_{main}$, resulting in significant pre-emphasis (bottom). Both pre- and post-tap time delays were set to $(3/16) \times UI$. (c) Measured eye diagram (PRBS-31) at 1.0 Gb/s using the same cable, weight vector $[5, 7, 5]$, and time delays of $(3/16) \times UI$.

N. St. John et al., *Journal of Instrumentation* (2023)

Backup slides

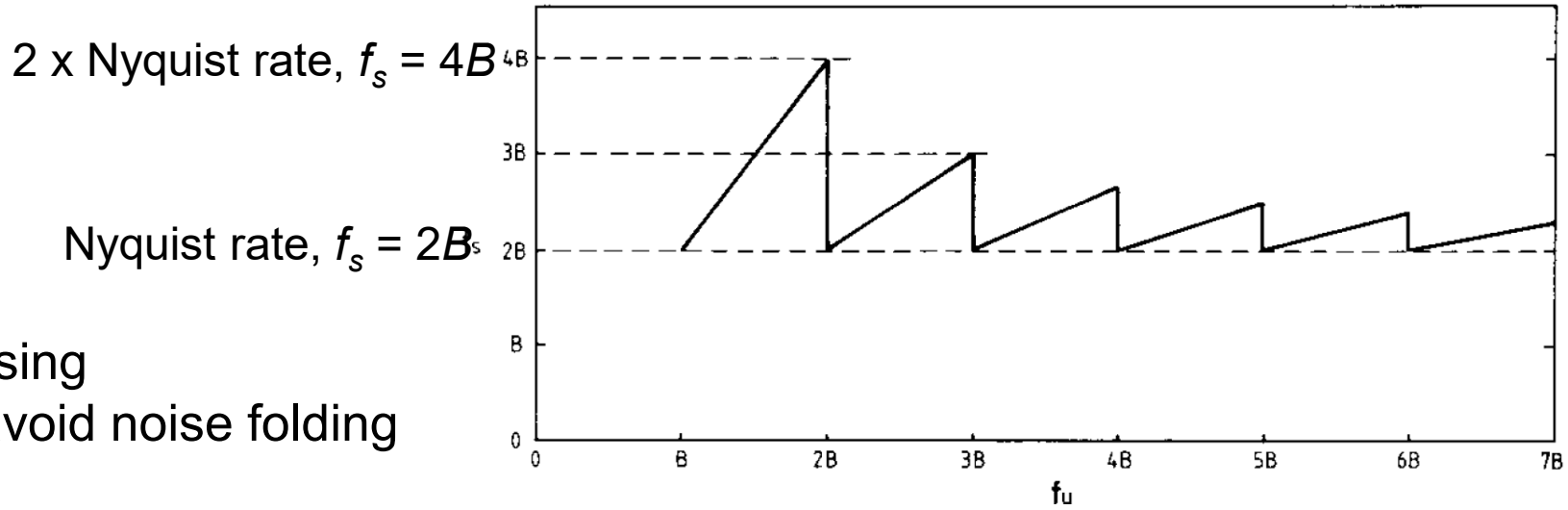
Subtitle

How do we sample RF signals?

Most RF signals are narrowband, i.e., are limited to a bandwidth of B around a center frequency f_0 where $B << f_0$. The most common way to sample them is to first remove f_0 by using a down-converting mixer.

Another way: The Nyquist sampling theorem requires that $f_s > 2B$ to avoid aliasing. Here B, the signal bandwidth, need not be centered around DC. In fact, the band-pass sampling theorem only requires that

$$f_s > \frac{f_u}{n}, \quad n = \left\lfloor \frac{f_u}{B} \right\rfloor \text{ for a signal between } f_l \text{ and } f_u = f_l + B.$$



- Avoids aliasing
- Does **not** avoid noise folding

Vaughan et al., "The theory of bandpass sampling," *IEEE Transactions on Signal Processing* (1991)