Fundamentals of Time-Interleaved ADCs

John P. Keane
Keysight Technologies, Inc.

February 16, 2020

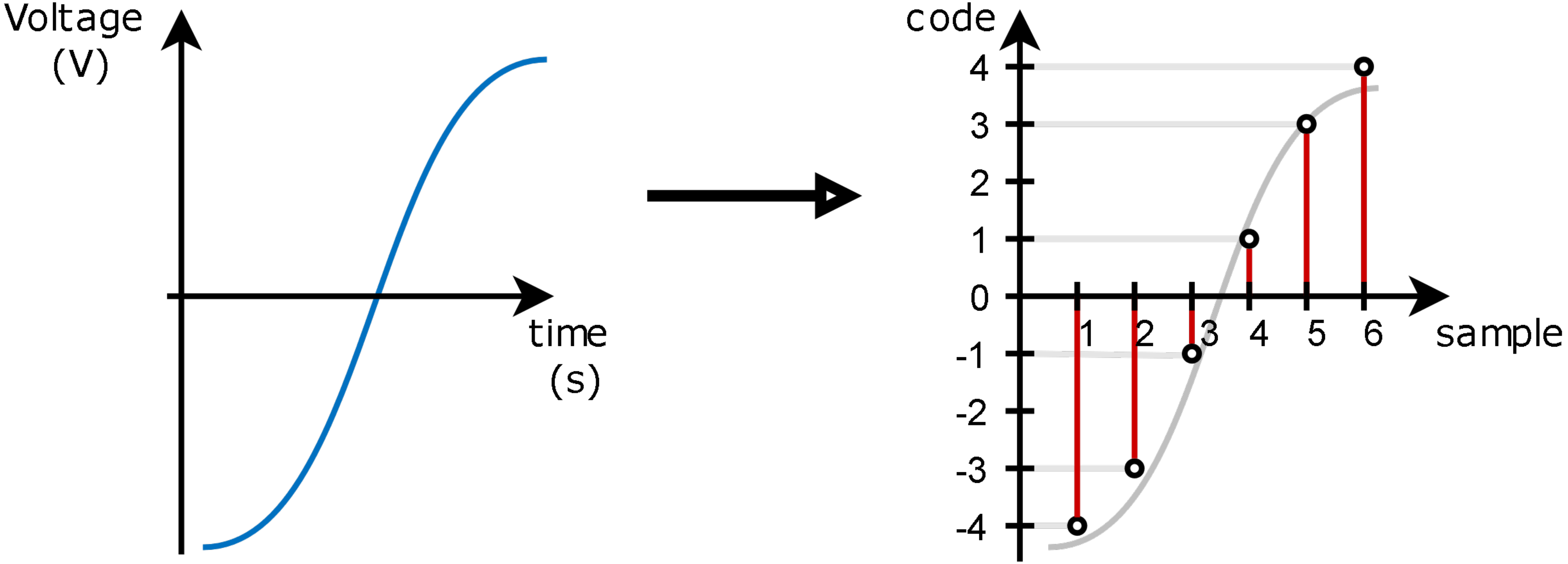
Abstract

- Abstract
  - In recent years, time-interleaving analog-to-digital converters (ADCs) has become more popular, especially for high sample rate applications such as wireline communications
  - This tutorial will cover the fundamentals of time-interleaved sampling, including an introduction to aliasing and an explanation of how mismatch in time-interleaved architectures can cause aliasing artifacts to appear
  - Practical methods to implement time-interleaved ADCs and combat these mismatch-induced effects will also be presented
- Motivation for time-interleaving
  - ADC design is a tradeoff between speed, accuracy and power
  - Time-interleaving gives an extra degree of freedom
    - Allows us to design a high-accuracy ADC with faster speed
    - Allows us to design a high-speed ADC with lower power

Outline

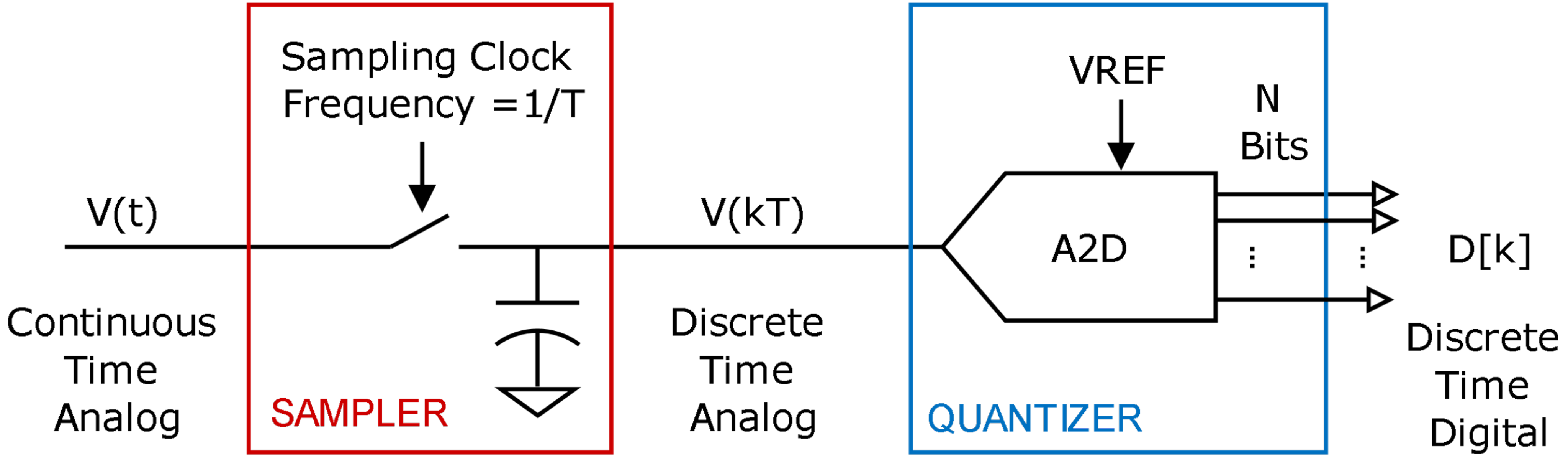
- Background and Motivation
- Error Sources in Time-Interleaved ADCs
- Interim Q+A
- Frequency Spectrum of Interleave Errors
- Correction of Interleave Errors
- Time-Interleaved ADC Architectures
- Summary and Conclusions

What is an Analog-to-Digital Converter



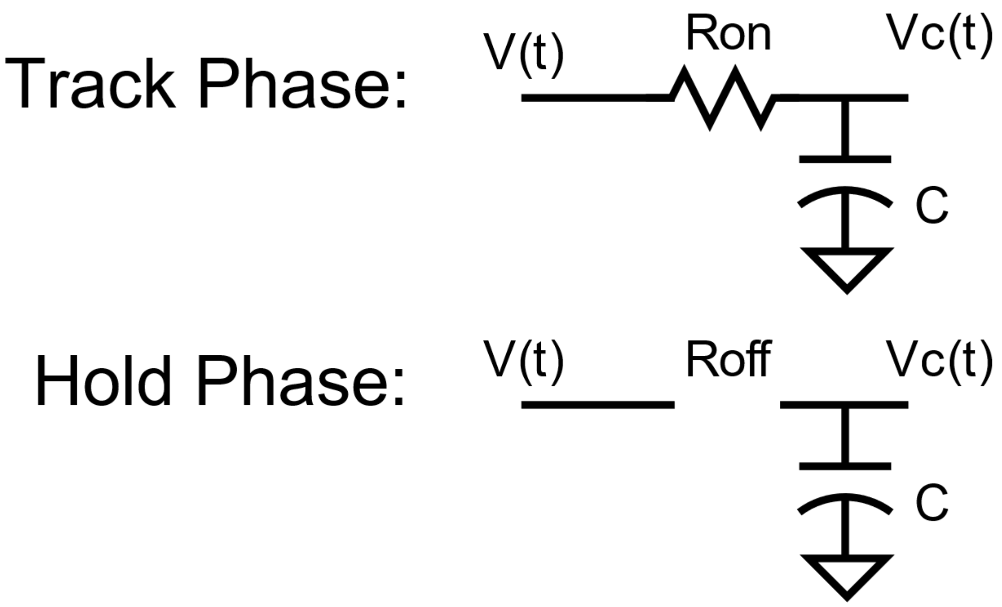
- Converts from an **Analog** signal to a **Digital** signal
  - Defined only at discrete points in time (**Sampled**)
  - Has only a discrete number of possible values (**Quantized**)

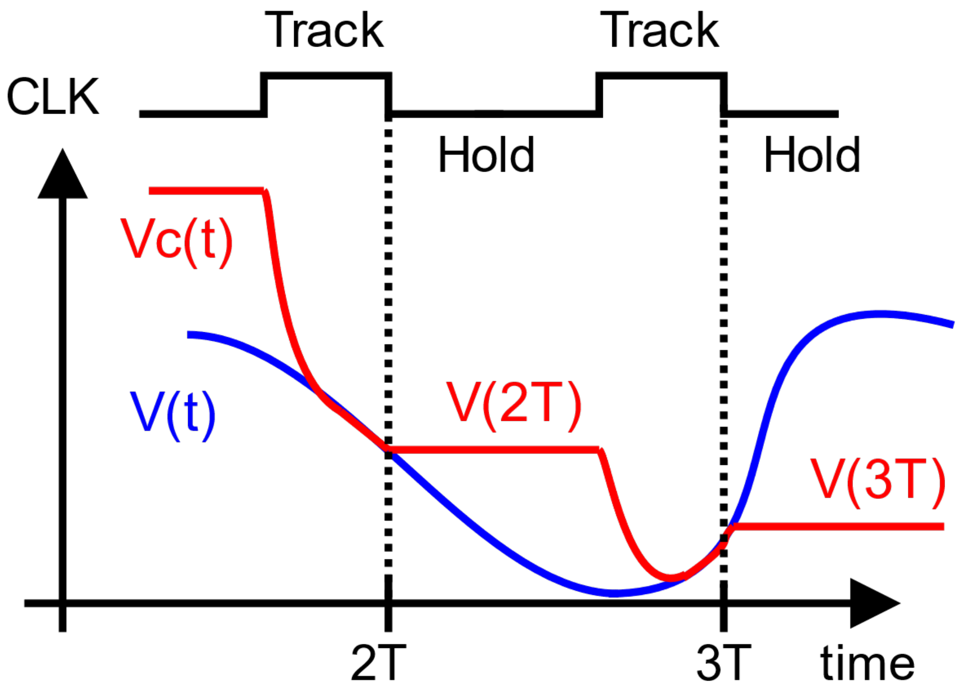
ADC Structure



- □ Sampler (Track-and-Hold)
  - Samples the continuous signal V(t) once per sample period T
  - Holds the voltage V(kT) until the next sample time
- □ Quantizer
  - Converts held voltage V(kT) to digital form D[k]
  - Calculates ratio of held voltage to reference voltage VREF => D[k] ≈ V(kT)/VREF

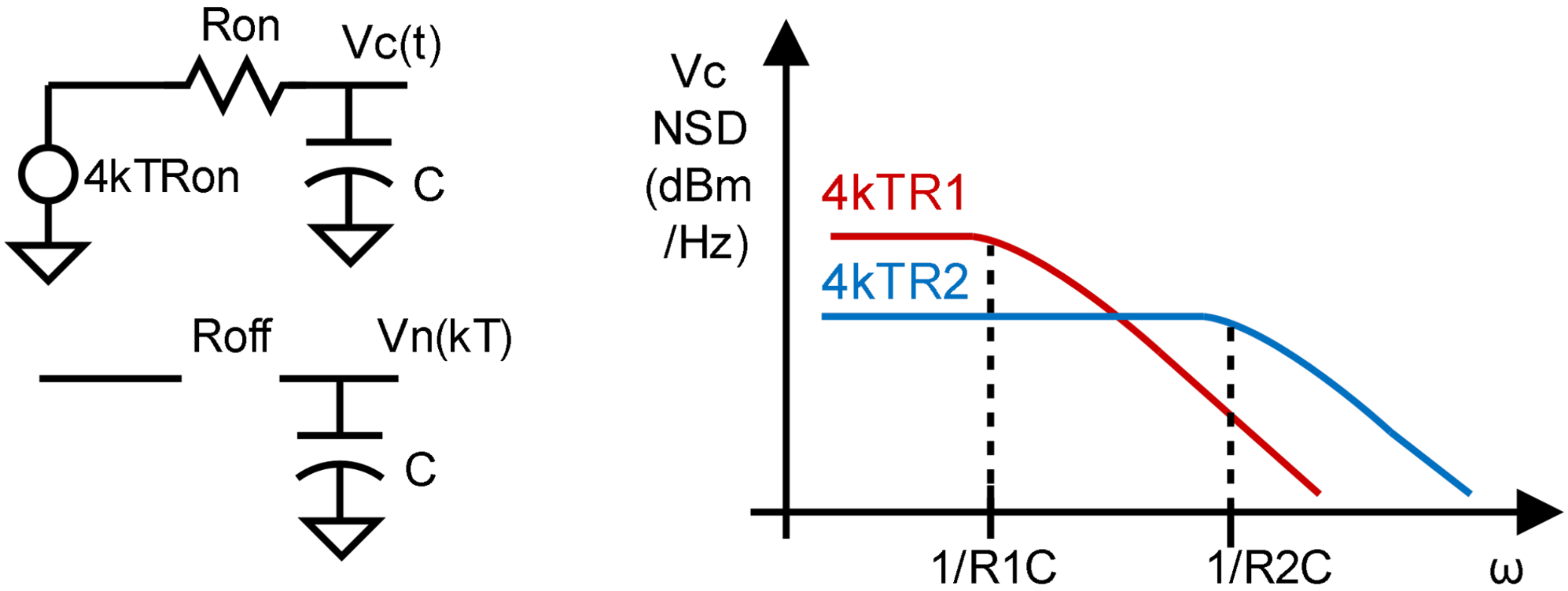
Track-and-Hold Sampler (T/H)





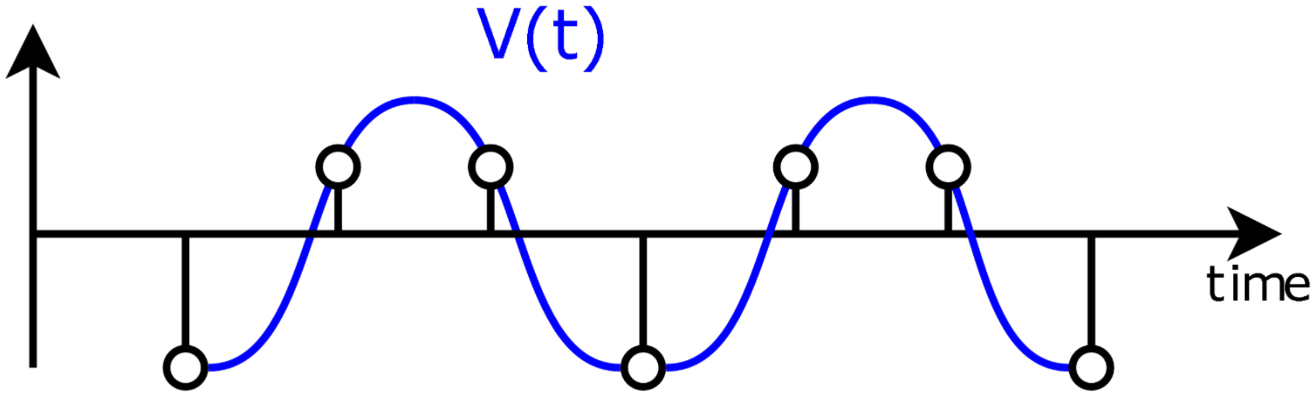
- □ Simple operation in principle
  - Close switch to **track** input – switch resistance is R_{on}
  - Open switch to **hold** sample – switch resistance is R_{off}
- □ Time constant = $R_{on}C$ => Fast if C is small and R_{on} is low

Sampling Noise of T/H



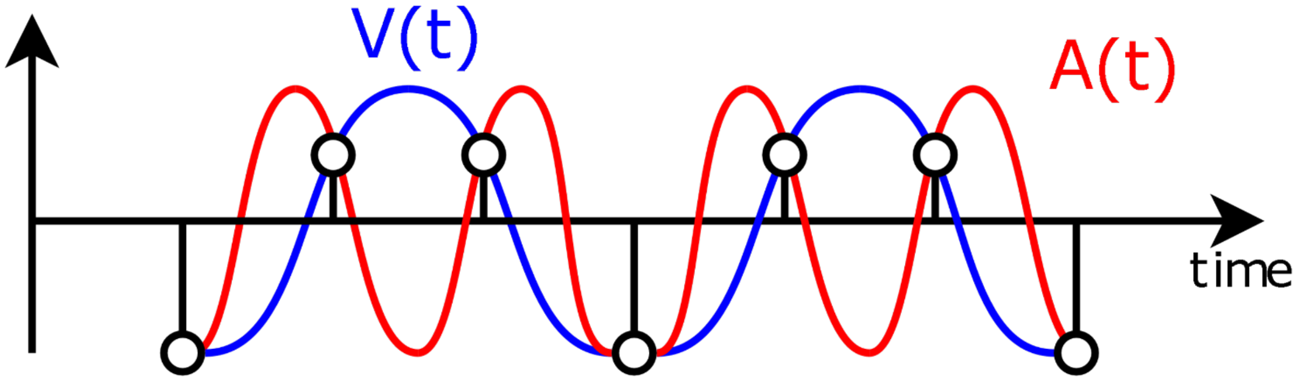
- □ Noise Spectral Density (NSD) of switch R_{on} = $4kTR_{on}$
  - ■ Large R_{on}=R1 => Higher NSD, Lower integrating bandwidth 1/R1C
  - ■ Small R_{on}=R2 => Lower NSD, Higher integrating bandwidth 1/R2C
- □ RMS value of sampled noise = $\sqrt{kT/C}$ independent of R_{on}

Sampling Theory



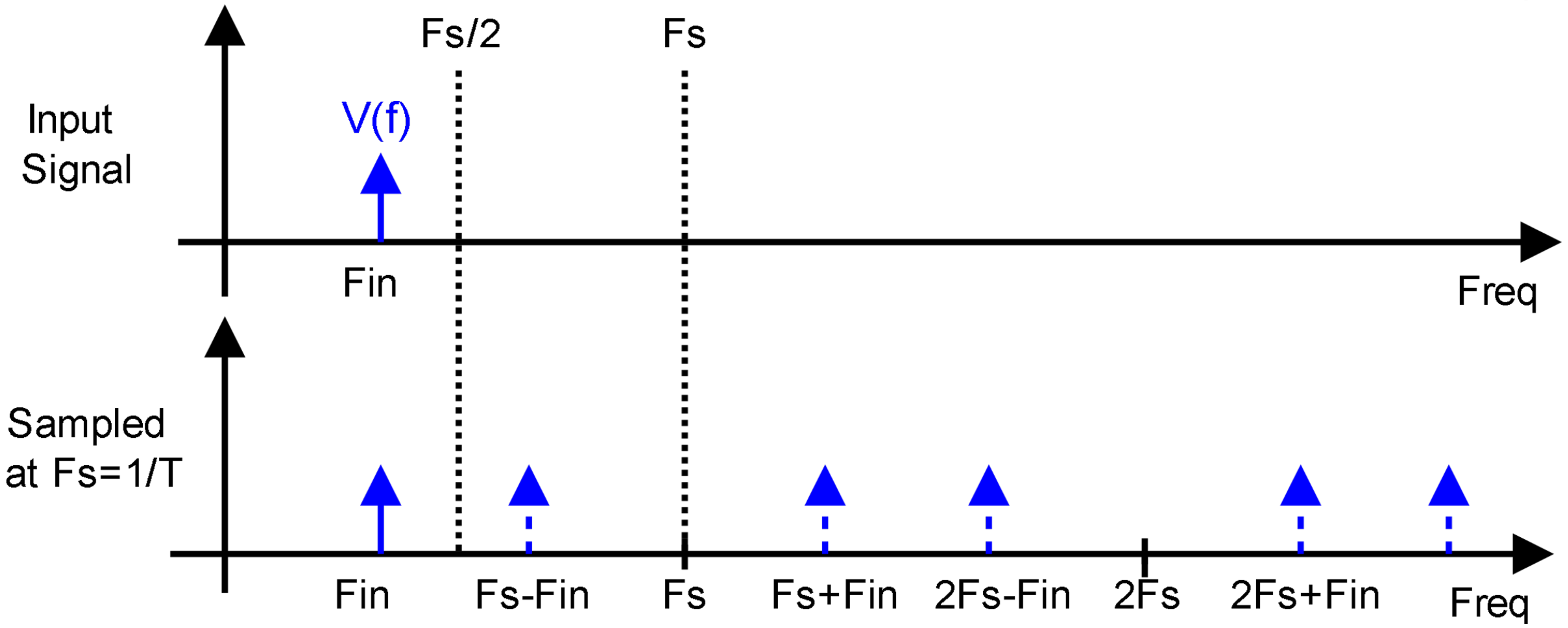
- □ Consider a signal V(t) sampled uniformly at Fs=1/T
- □ Can we reconstruct V(t) from just the samples V(kT) ?
  - Nyquist theory says we can!
- □ But is this the only solution?

Aliasing



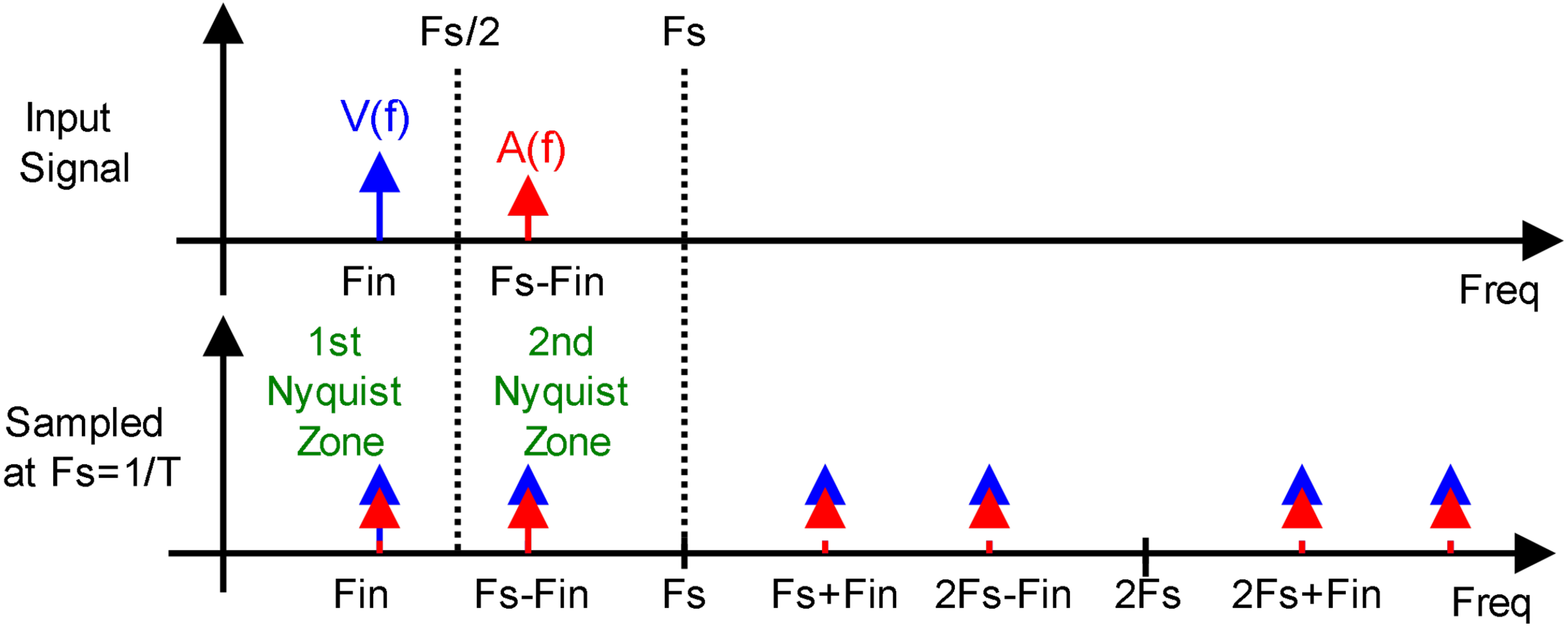
- □ No, A(t) is also a solution!
- □ Set of samples V(kT) coincide with both V(t) and A(t)
  - ■ V(t) and A(t) are both *Aliases* of each other
- □ No way to tell which was the original waveform
  - ■ Need another piece of information

Aliasing in the Frequency Domain



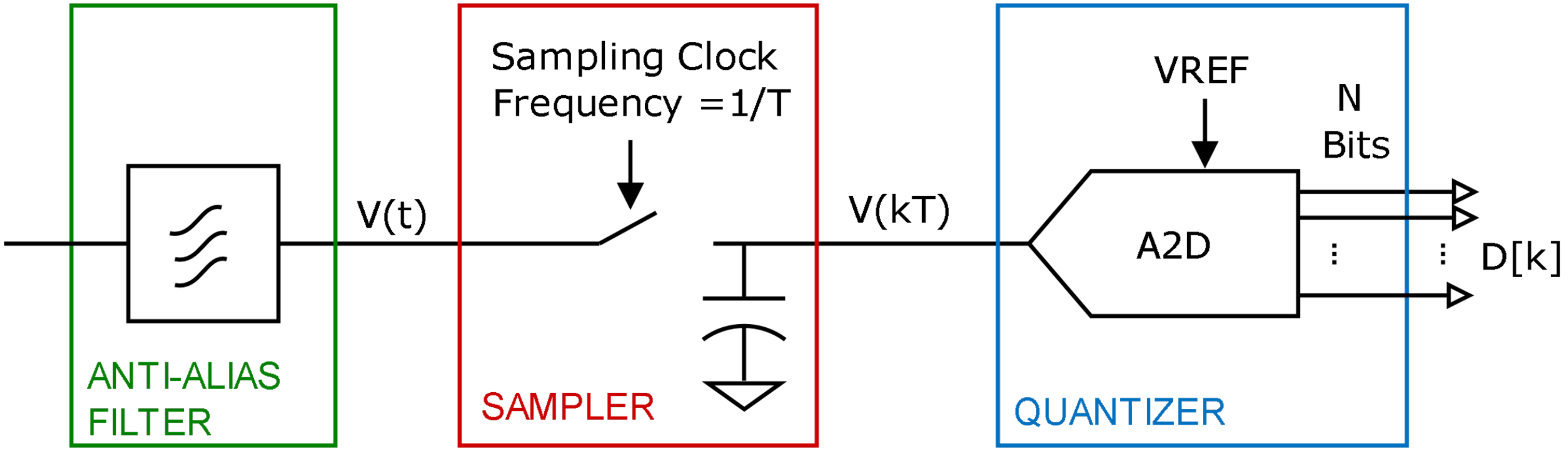
- □ Input signal V(t) is a sinewave of frequency Fin<Fs/2
- □ Aliases Occur at N·Fs ± Fin for every integer N

Aliasing in the Frequency Domain



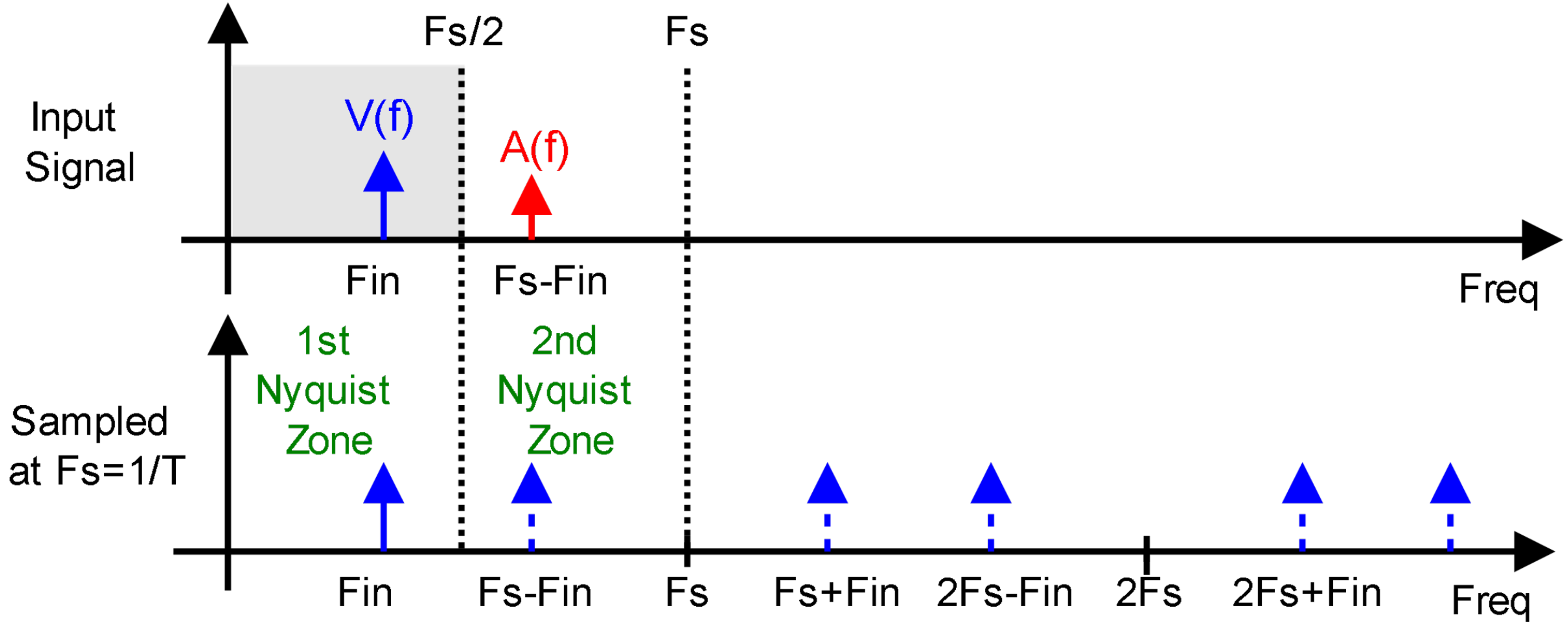
- □ What if input signal has component A(f) at Frequency Fs-Fin?
- □ Aliases of V(f) and A(f) coincide – cannot distinguish

Anti-Alias Filter



- □ Real ADC systems always include an anti-alias filter
  - ■ Or some way the input signal is limited to a single Nyquist zone
- □ E.g. a passive filter on a board, or an active filter on-chip

Low-pass Anti-Aliasing Filter

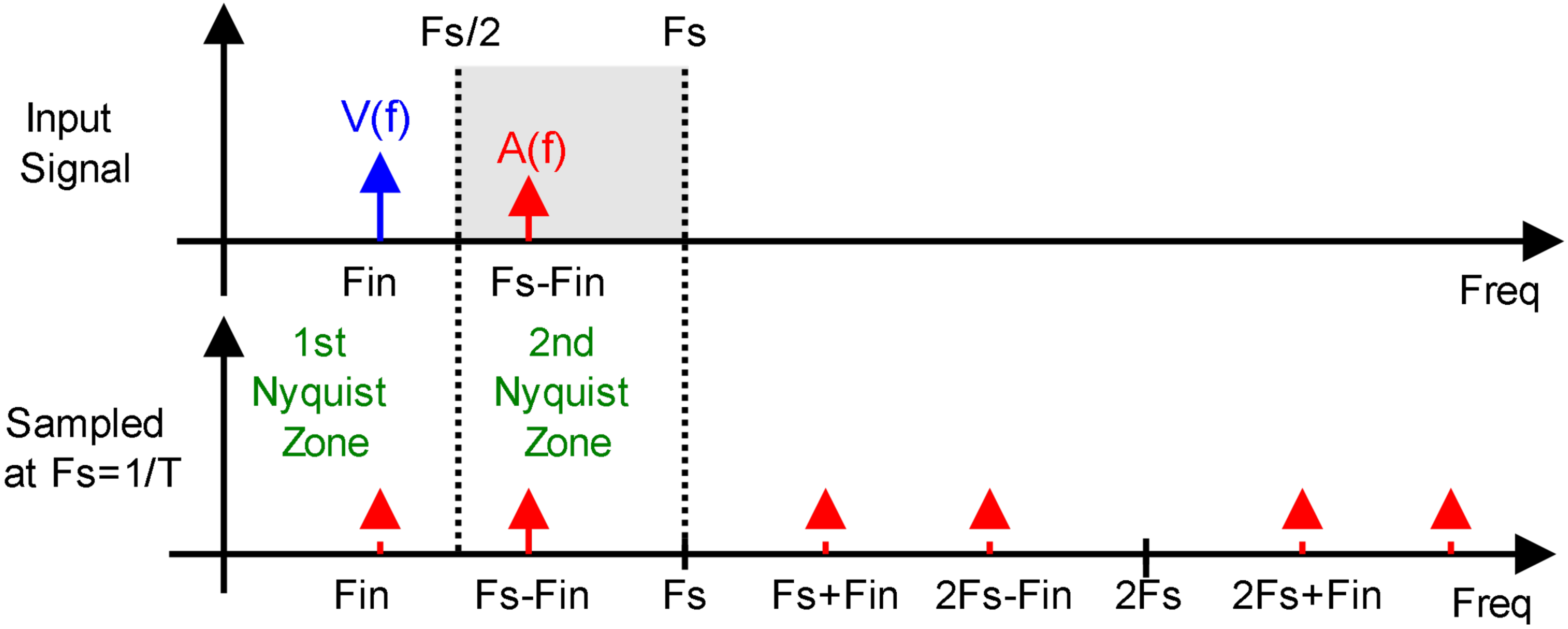


□ Add a Low-pass Anti-Alias filter (AAF) before the sampler

- Selects 1st Nyquist zone by suppressing all frequencies > Fs/2

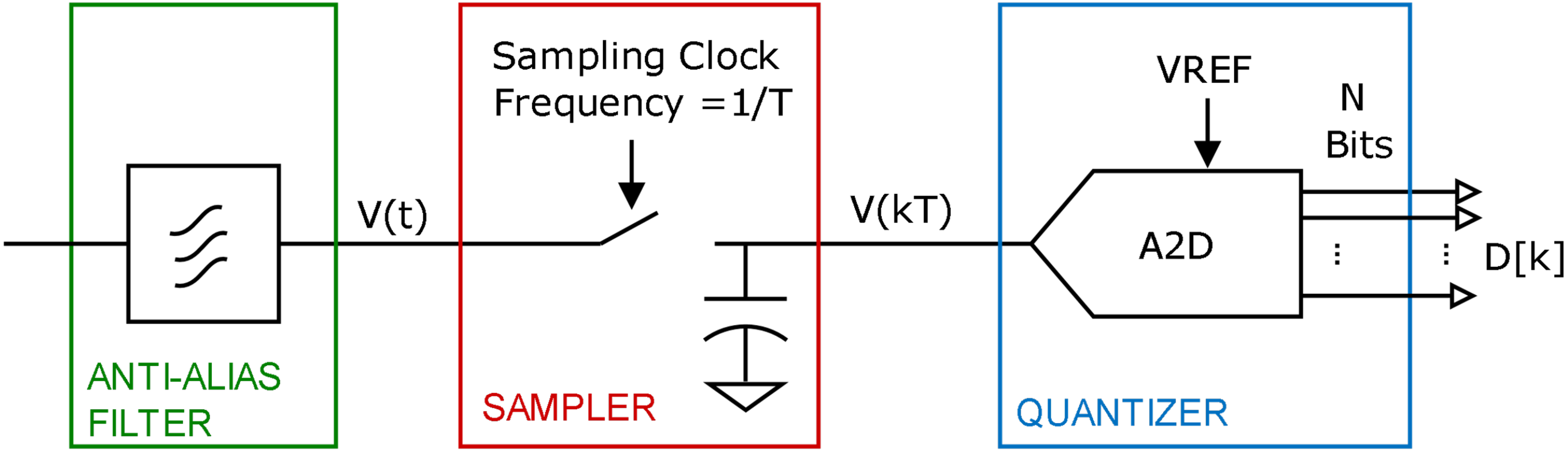
□ We can reconstruct the input signal V(t) without A(f)

Band-pass Anti-Aliasing Filter



- □ But what if we wanted A(f) not V(f) ?
- □ Add a Band-pass Anti-Alias filter before the sampler
  - Selects 2nd Nyquist Zone by suppressing all frequencies > Fs/2 and <Fs

Quantizer



- □ Converts held voltage V(kT) to N-bit digital code D[k]
  - Compare V(kT) to VREF so D[k] ≈ V(kT)/VREF
  - Quantizer Error Q(kT) = D[k]·VREF – V(kT)
- □ Many different architectures
  - Choice depends on requirements for resolution, power, speed, latency

Quantizer Resolution

- Number of bits (N)
  - Literally, the number of bits in the output code of the quantizer
  - Measure of *precision*, not accuracy
- Signal-to-Noise and Distortion Ratio (SNDR)

$$SNDR = \log_{10} \frac{Signal\ Power}{Noise + Distortion\ Power} = \log_{10} \frac{E[V^2(kT)]}{E[Q^2(kT)]}$$

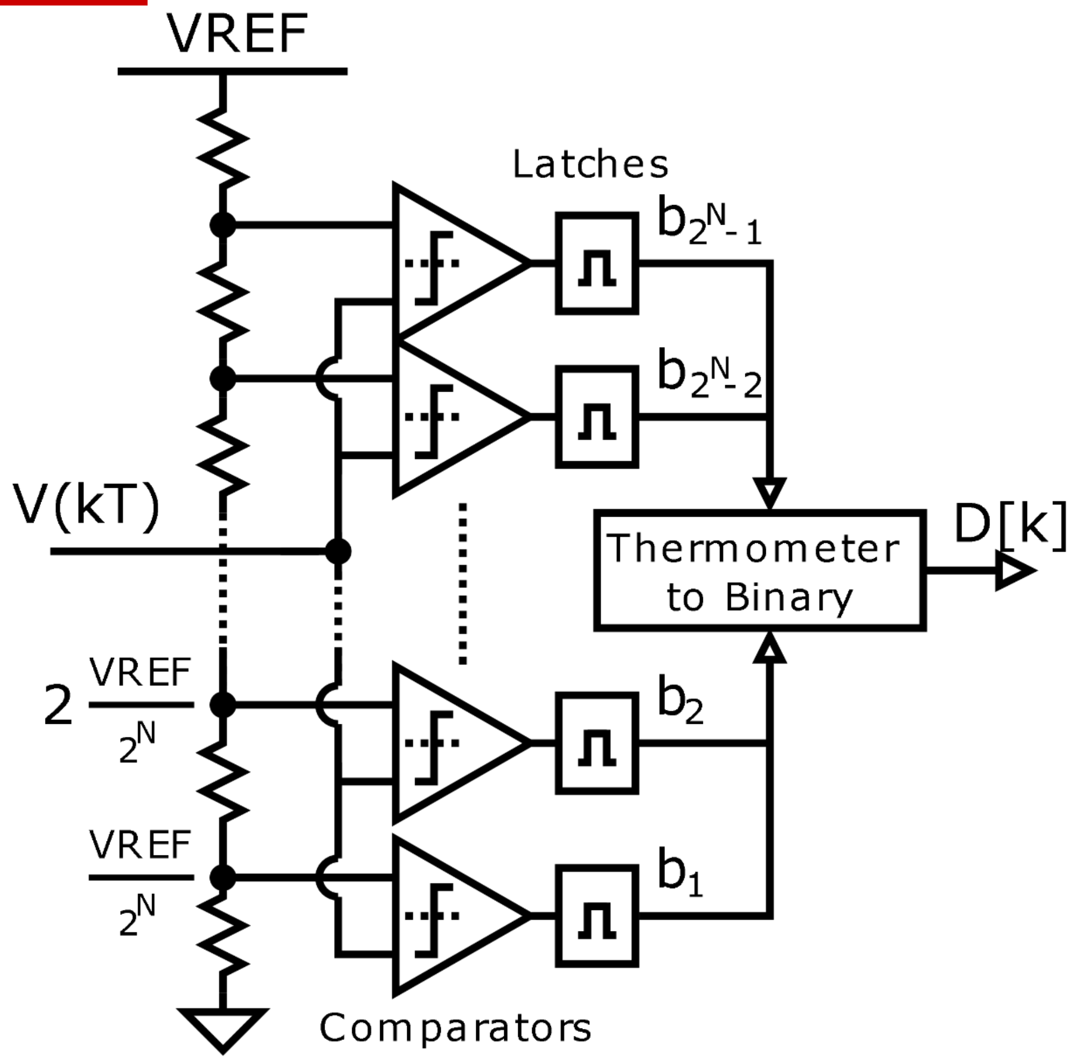
  - Measure of *accuracy* of the conversion
- Effective Number of Bits (ENOB)

$$ENOB = \frac{SNDR - 1.76}{6.02}$$

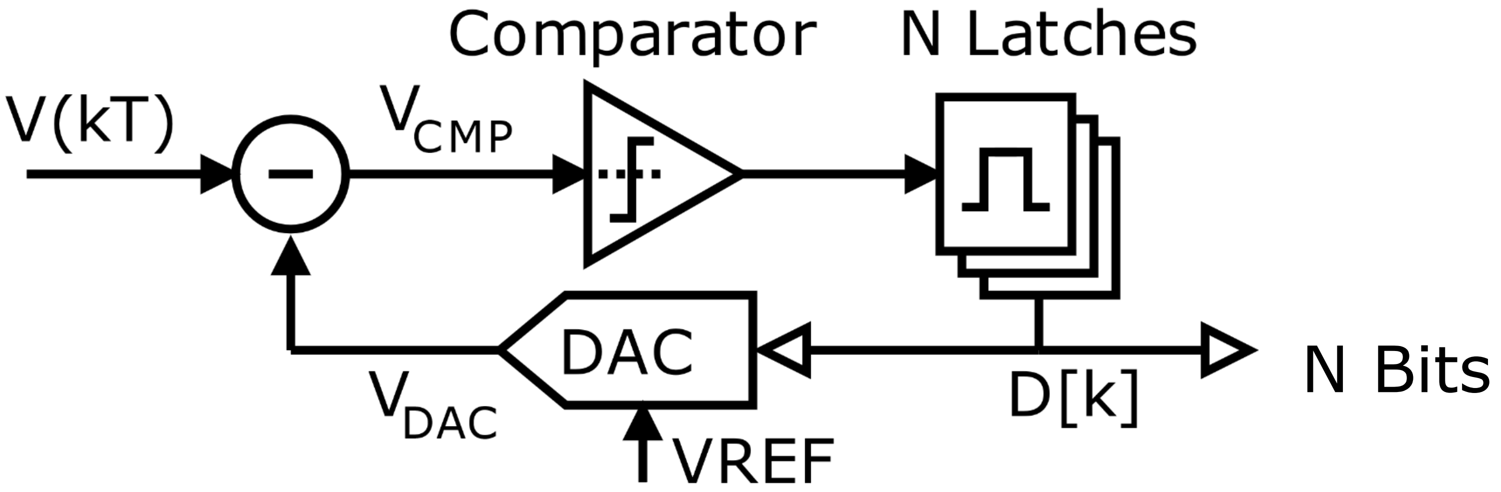
  - ENOB is number of bits an *ideal* quantizer needs for this SNDR
  - ENOB ≤ N for all non-ideal quantizers

Flash ADC

- □ Compares to 2^N -1 levels at once
  - Stores 2^N -1 decisions b_n in latches
  - Converts to N-bit binary code D[k]
- □ Single operation
  - Very fast
- □ Requires 2^N -1 comparators
  - 63 comparators for 6-bit ADC
  - 1023 comparators for 10-bit ADC
- □ Good for 4-6 bit resolution
- □ Too much power, area and input load for >7 bits

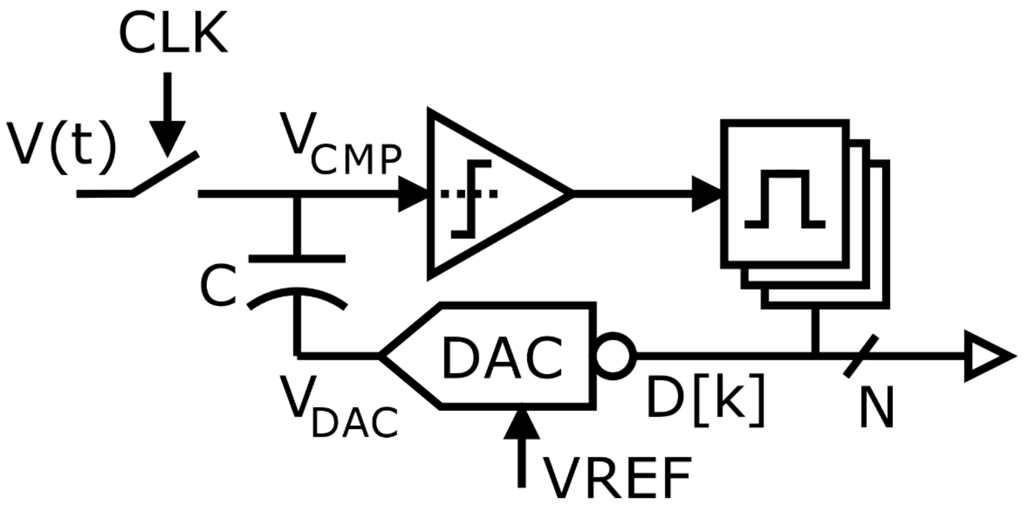


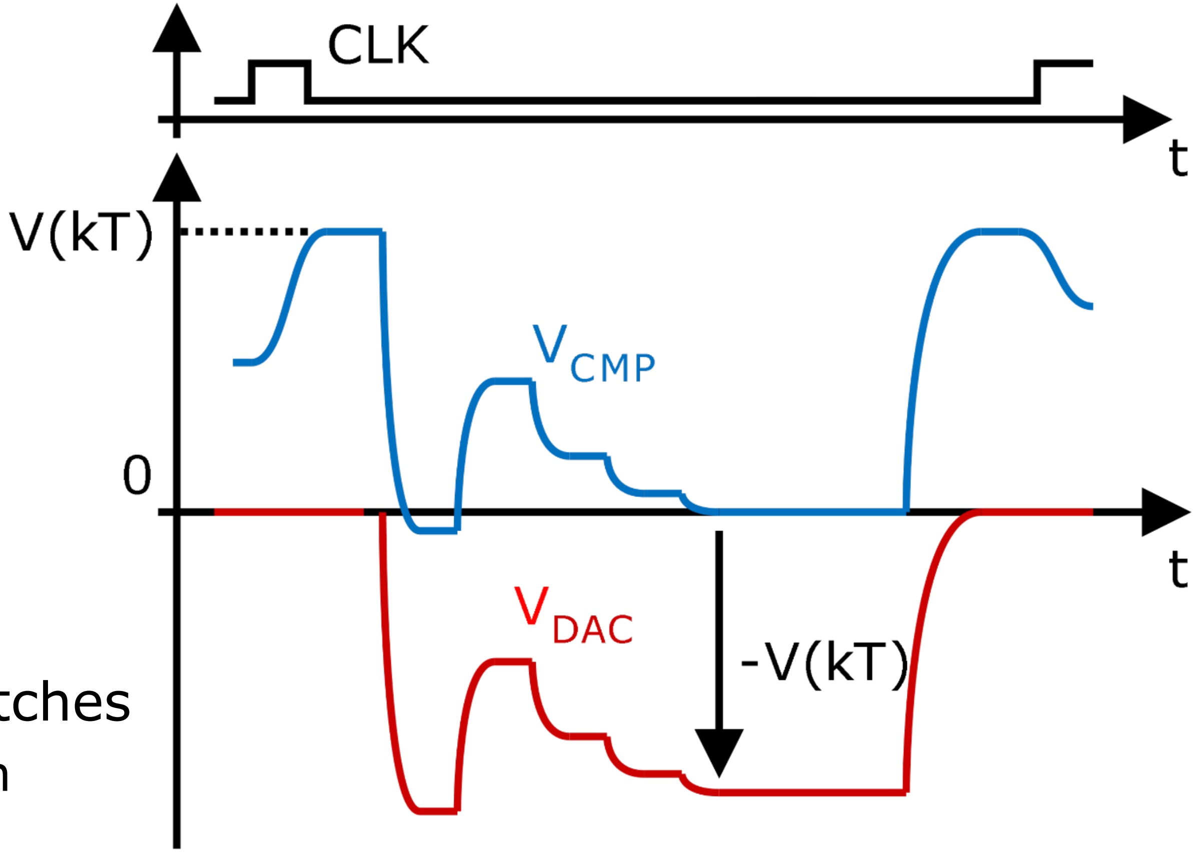
SAR (Successive Approximation Register)



- □ Approximate V(kT) with a DAC output V_{DAC}
- □ Use Successive Approximation algorithm
  - Comparator generates bits of D[k] one at a time, starting with MSB
  - Successively drives $V_{CMP} \approx 0$ so $V_{DAC} \approx V(kT)$
- □ DAC output $V_{DAC} \approx D[k] \cdot VREF$
  - $D[k] \approx V(kT)/VREF$ at end of conversion

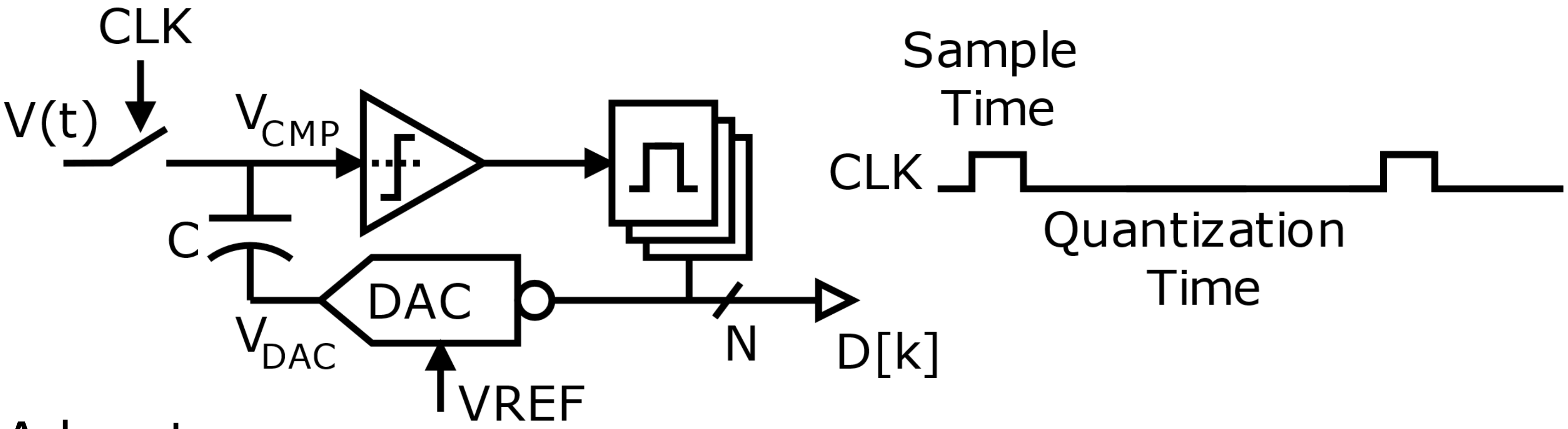
SAR ADC Implementation





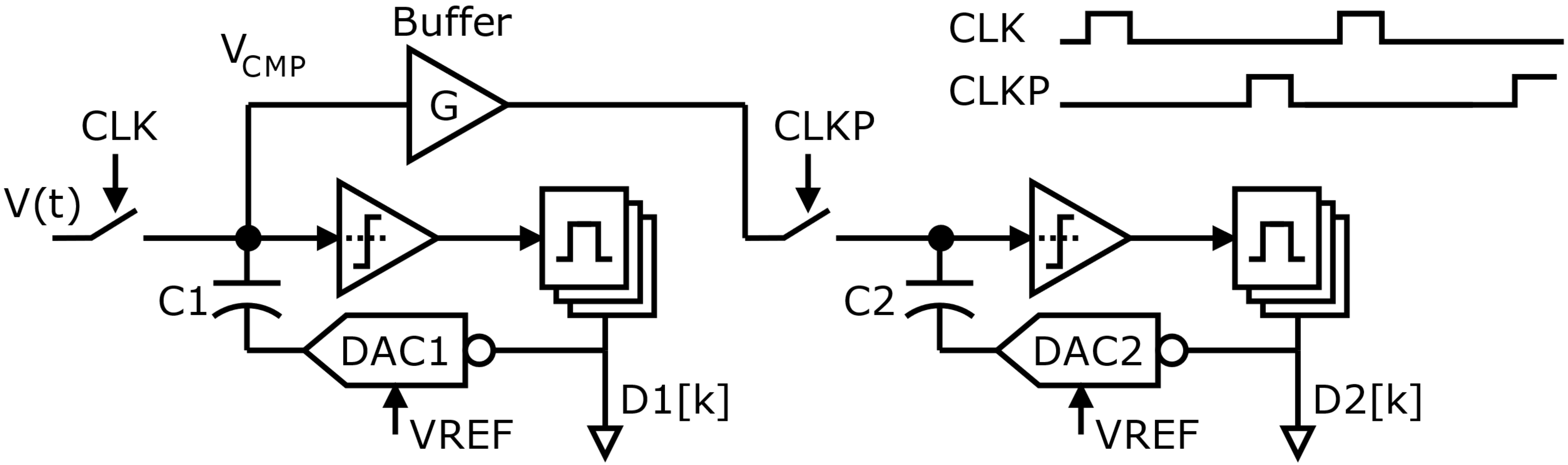
- □ Initialize D=0 so $V_{DAC}=0$
- □ Sample V(t) so V(kT) stored on C
- □ Store comparator decisions D in latches
  - ■ Forces $V_{CMP}\approx 0$ at end of conversion
  - ■ $V_{CMP} \approx 0 \Rightarrow V_{DAC} \approx -V(kT)$
  - ■ Invert D[k] => $D[k] \approx V(kT)/VREF$

SAR ADC Properties



- Advantages
  - Small, low power, integrated T/H
  - Scales well in modern CMOS processes
- Disadvantages
  - Takes many comparisons in series => slow
  - Quantization time much longer than sample time

Pipelined SAR ADC

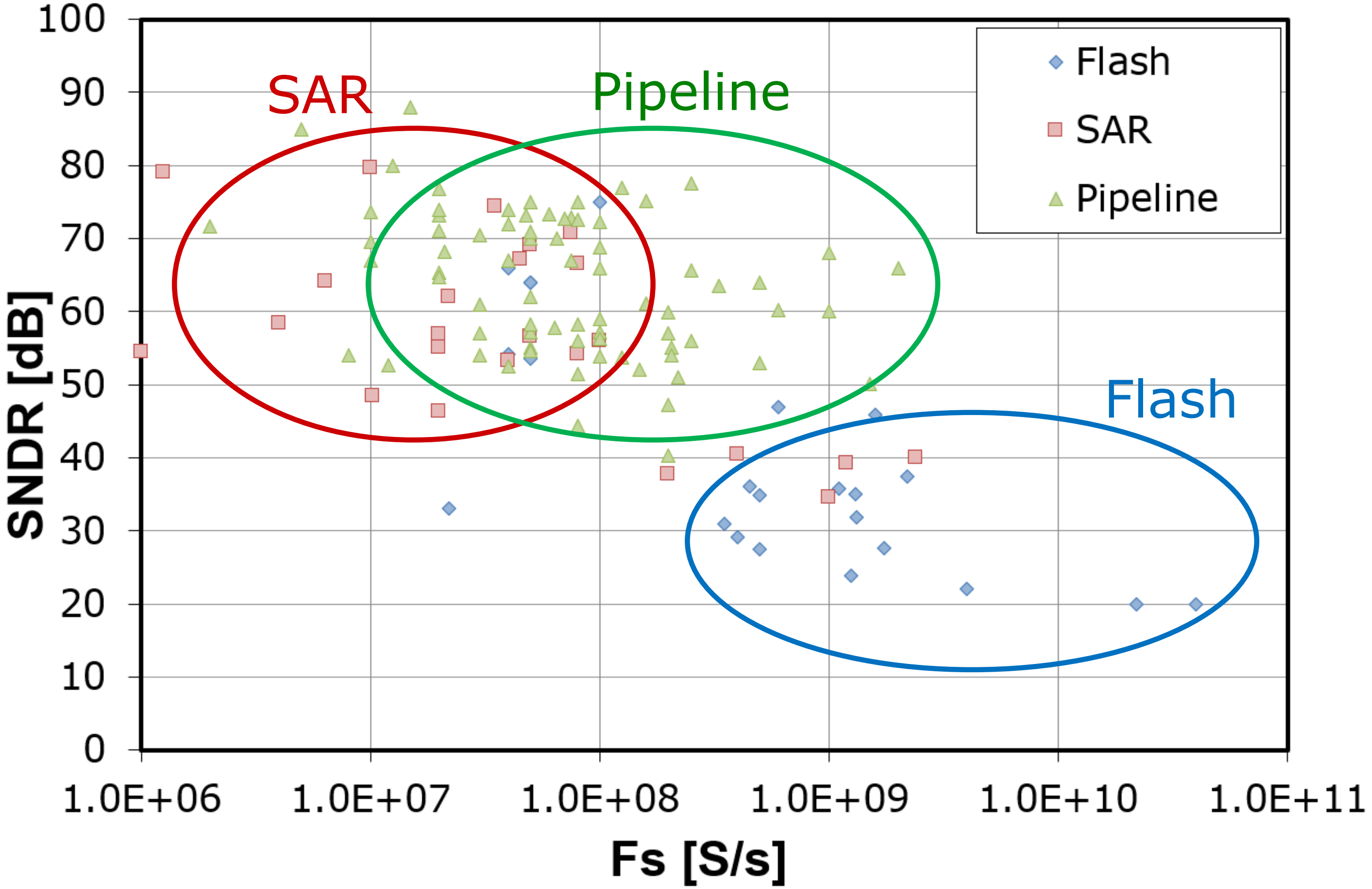


- □ SAR operation pipelined to improve speed
  - First stage computes MSBs D1[k], second stage computes LSBs D2[k]
  - Intermediate V_{CMP} from first stage sampled by second stage
- □ Fewer comparisons needed in first stage => Faster
- □ Buffer and additional T/H needed => More power and area

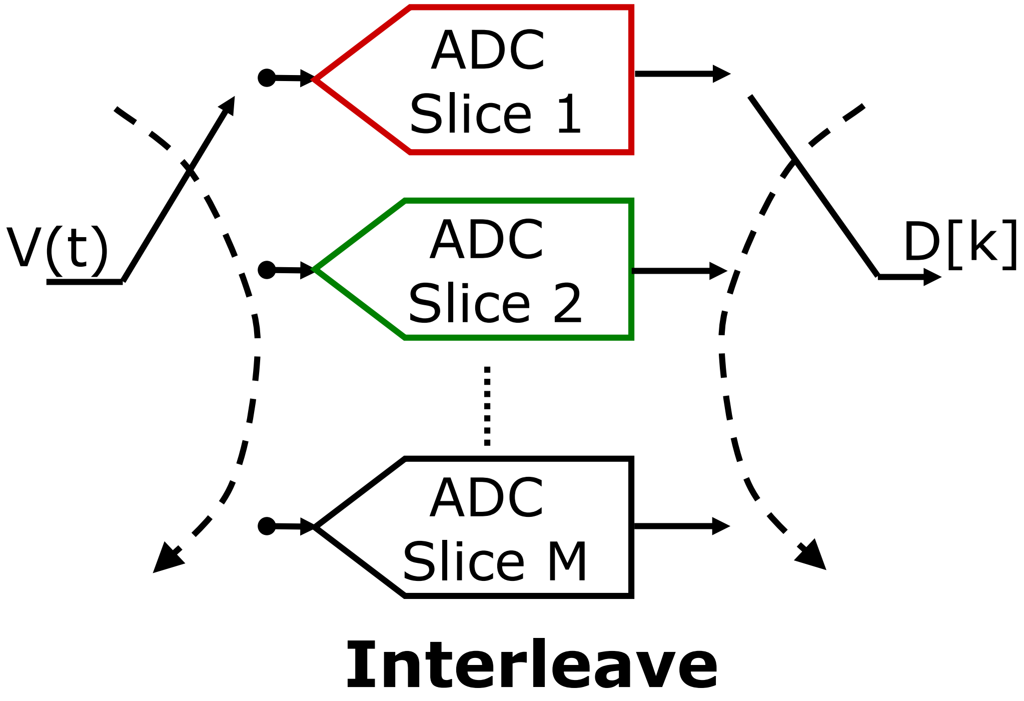
Performance Comparison SNDR vs. Fs

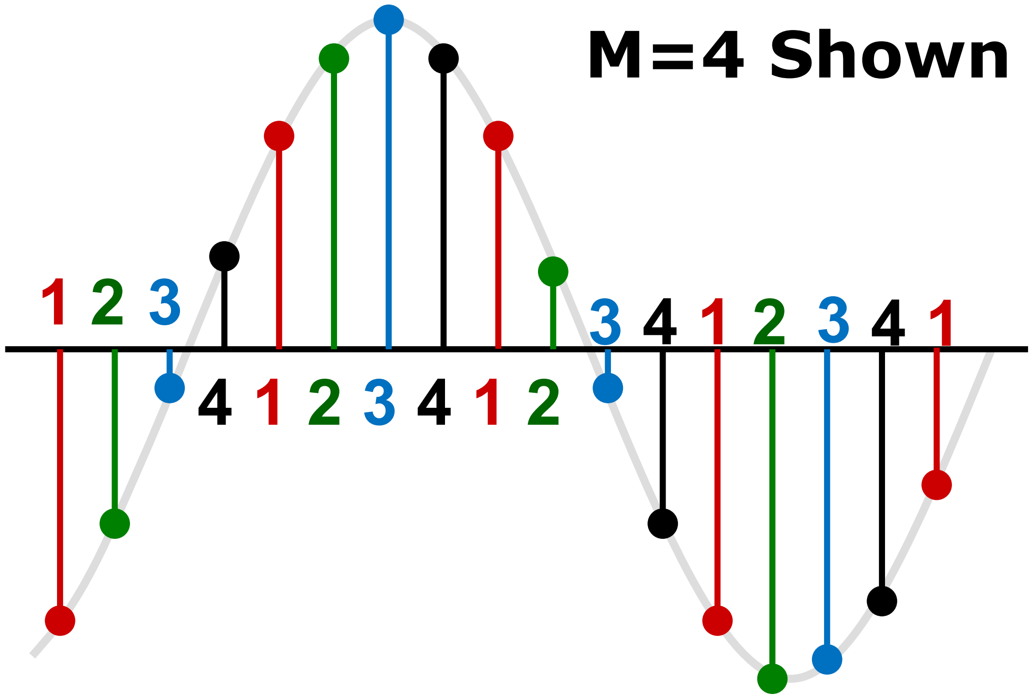
- □ SAR
  - ■ High SNDR
  - ■ Low sample rate
- □ Pipeline
  - ■ High SNDR
  - ■ Mid Sample rate
- □ Flash
  - ■ Low SNDR
  - ■ High sample rate

ISSCC 1996-2019 [1]



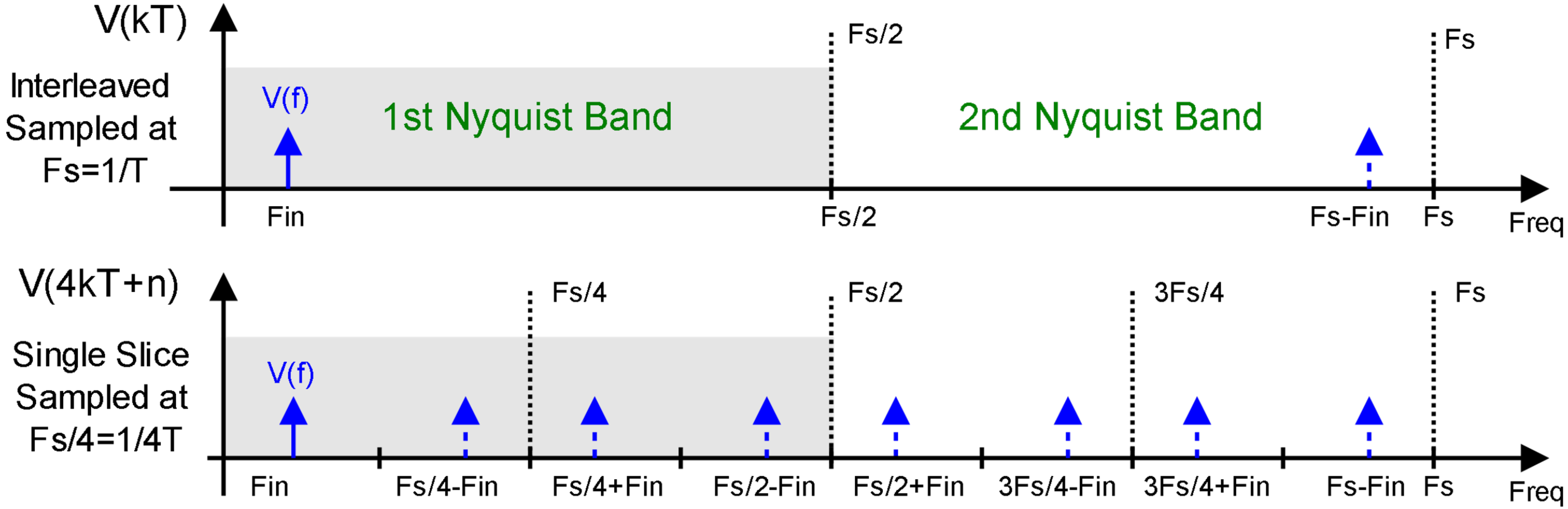
Time-Interleaved ADC [2]





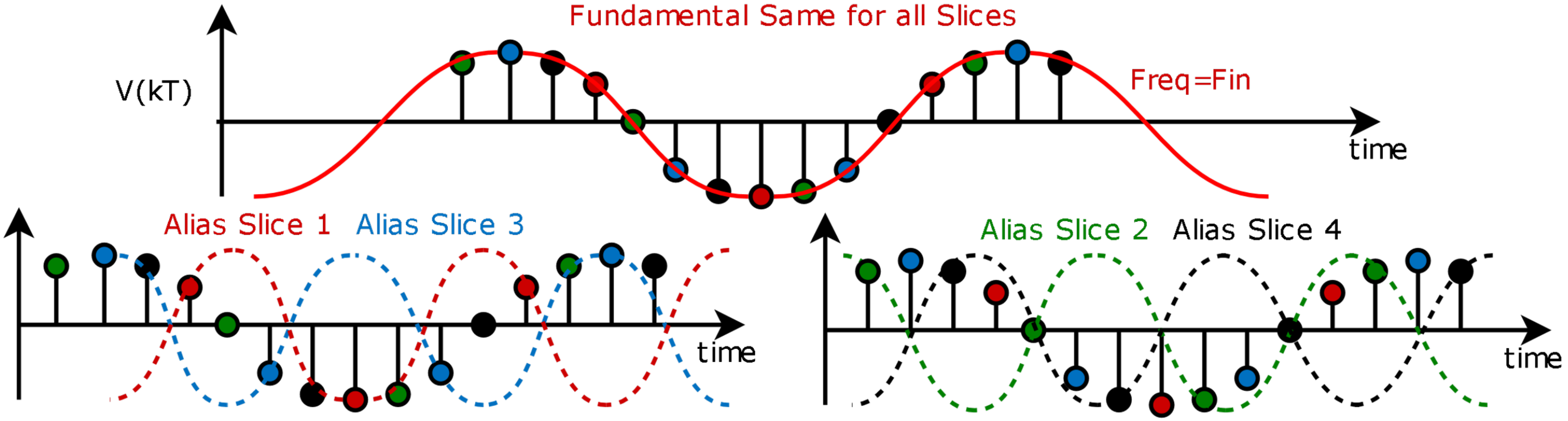
- □ Combine M identical ADC slices, to get M x Fs/M sampling.
  - Interleave input samples in Analog Domain
  - Interleave output samples in Digital Domain

Aliasing in ADC Slice with M=4



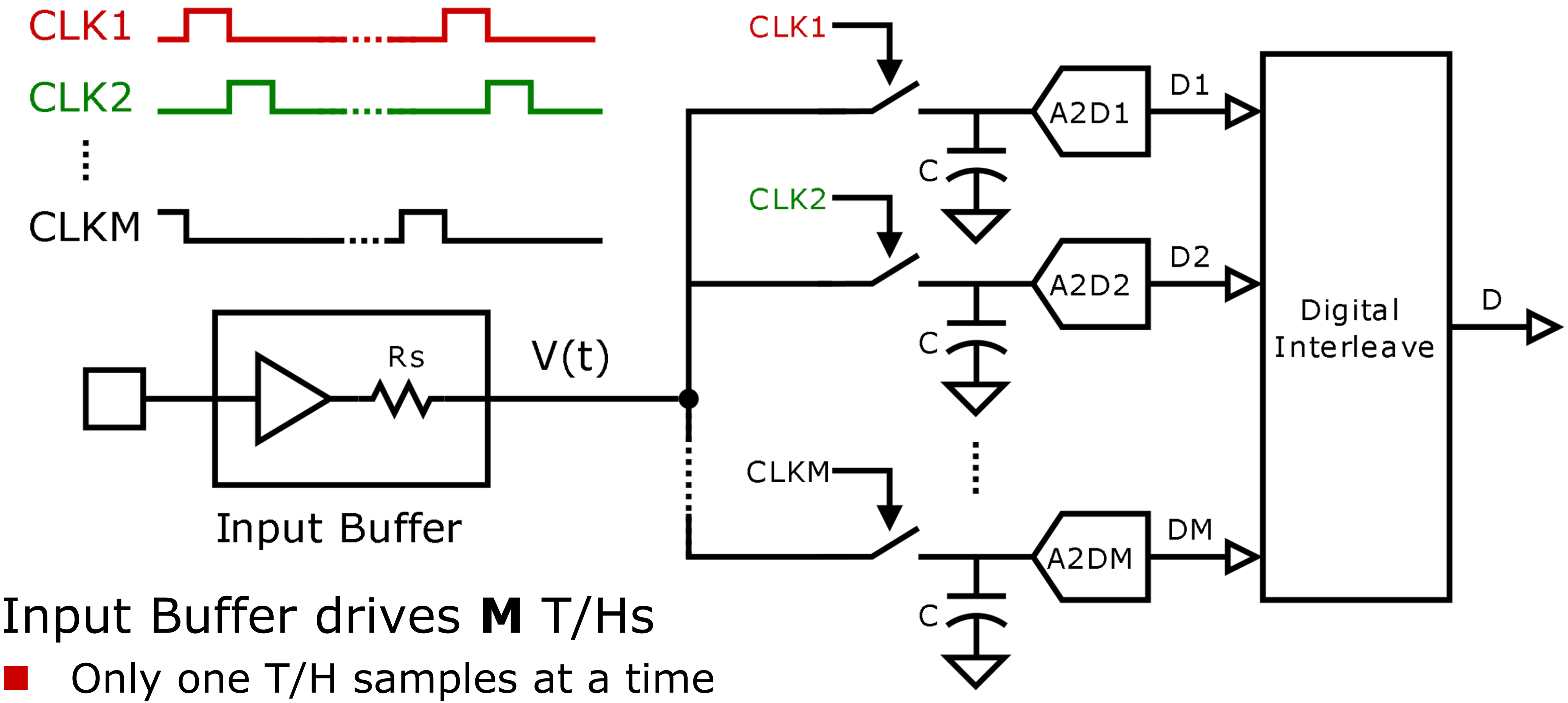
- □ Anti-Alias Filter (AAF) selects 1st Nyquist zone for full sample rate Fs=1/T
- □ ADC slice n input V(4kT+n) is V(t) sampled at Fs/4=1/4T
- □ Aliases occur at N·Fs/4±Fin => 3 alias terms in 1st Nyquist band

Summing Interleaved Aliases



- □ Each ADC Slice samples the input signal at Freq=Fin
  - ■ Fundamental at Freq=Fin has same phase for all slices
- □ Since it samples at Fs/4, each ADC Slice also sees an alias at Freq=Fs/4-Fin
- □ Alias has same frequency for each slice but **different phase**
  - ■ Alias terms sum to zero if all slices match exactly

Time-Interleaved ADC Implementation

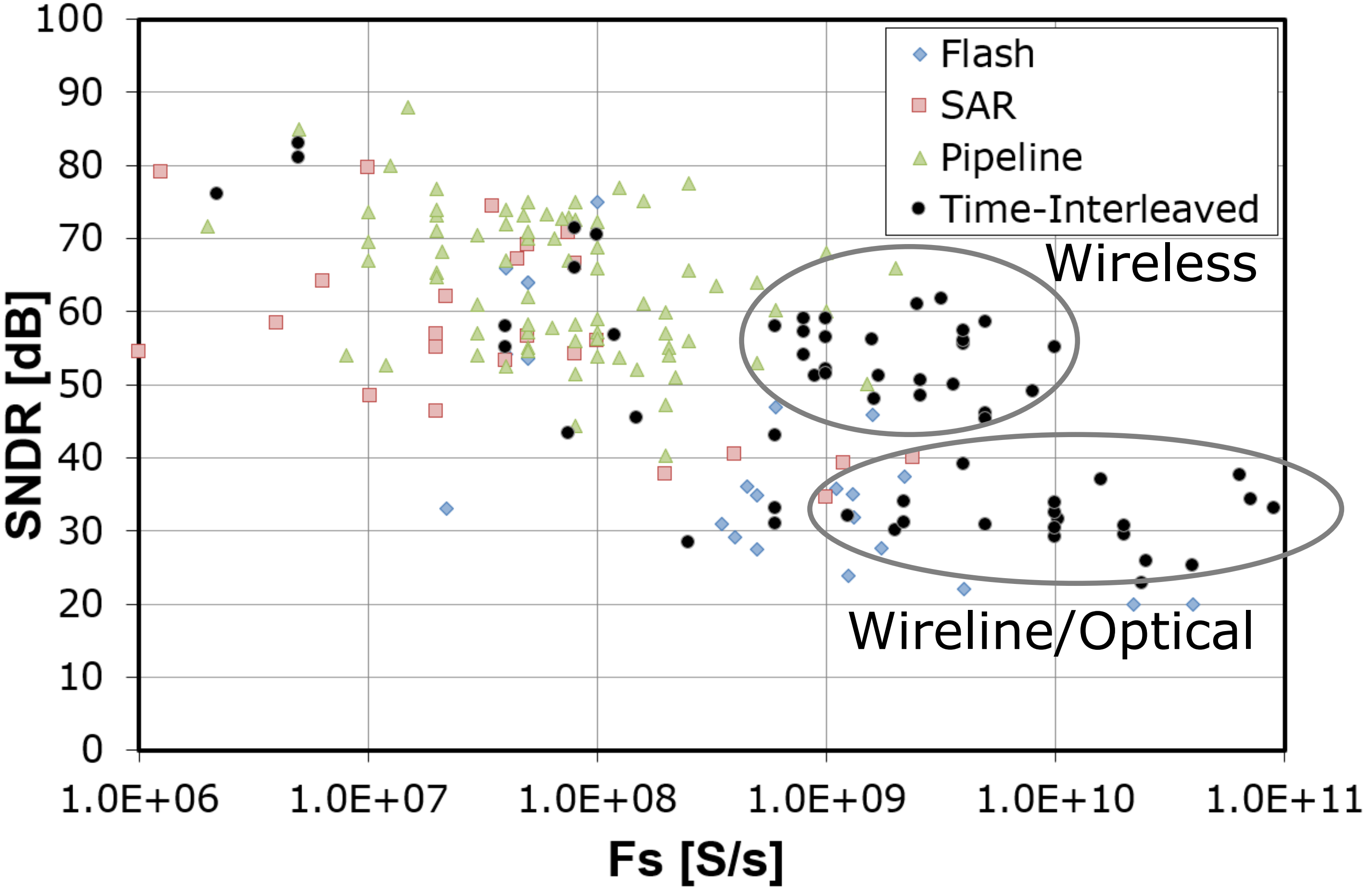


- □ Input Buffer drives **M** T/Hs
  - Only one T/H samples at a time
  - Re-interleave Quantizer outputs digitally

Time-Interleaved Applications

- □ Wireless
  - ■ >10b at 1-10GS/s
  - ■ Interleave M<10
  - ■ E.g. Base-station
    - □ 5GS/s 12-bit
- □ Wireline / Optical
  - ■ <7b at 1-100GS/s
  - ■ Interleave M>20
  - ■ E.g. 56Gbps PAM4
    - □ 28GS/s 6-bit

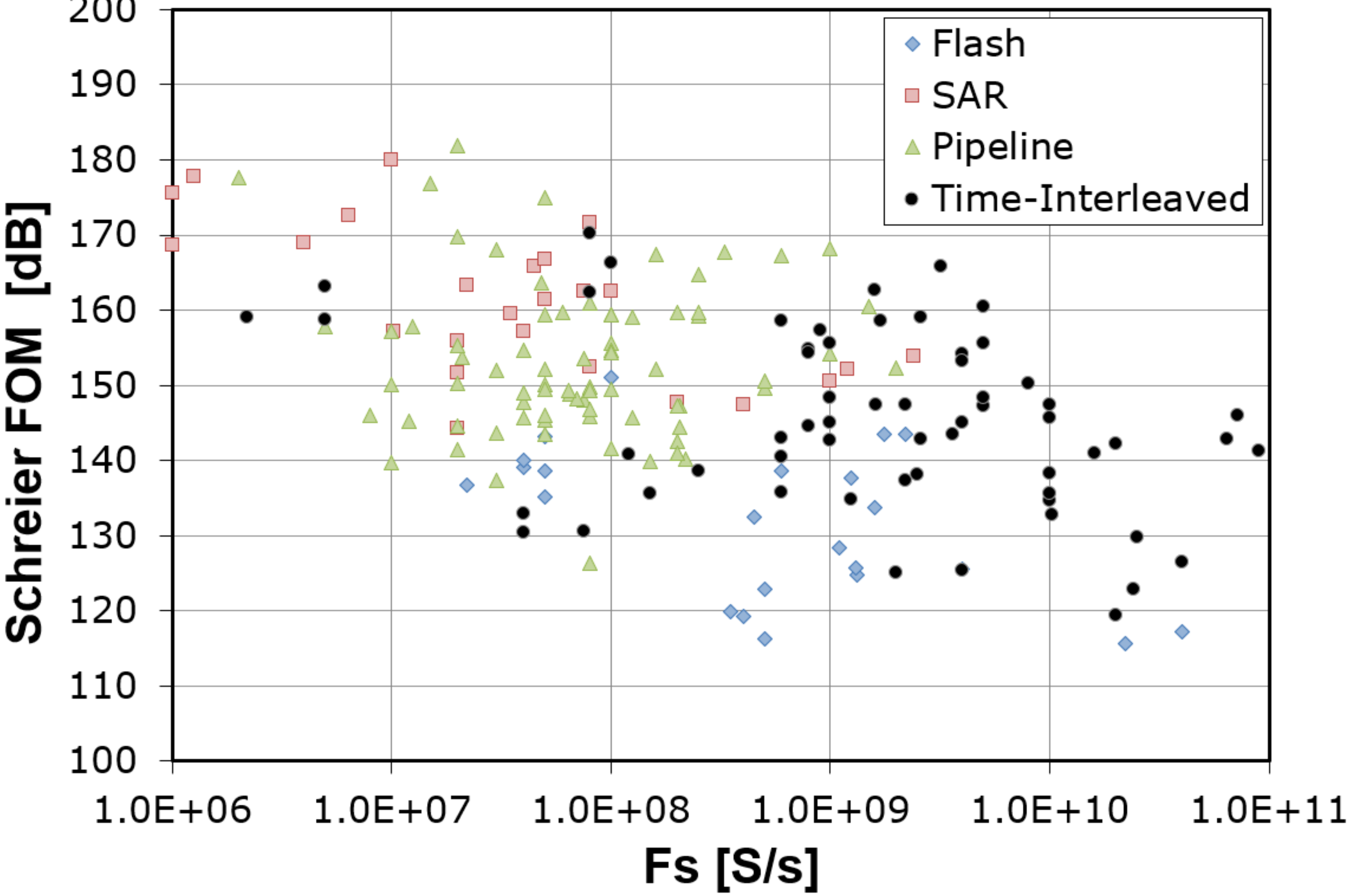
ISSCC 1996-2019 [1]



Why Interleave?

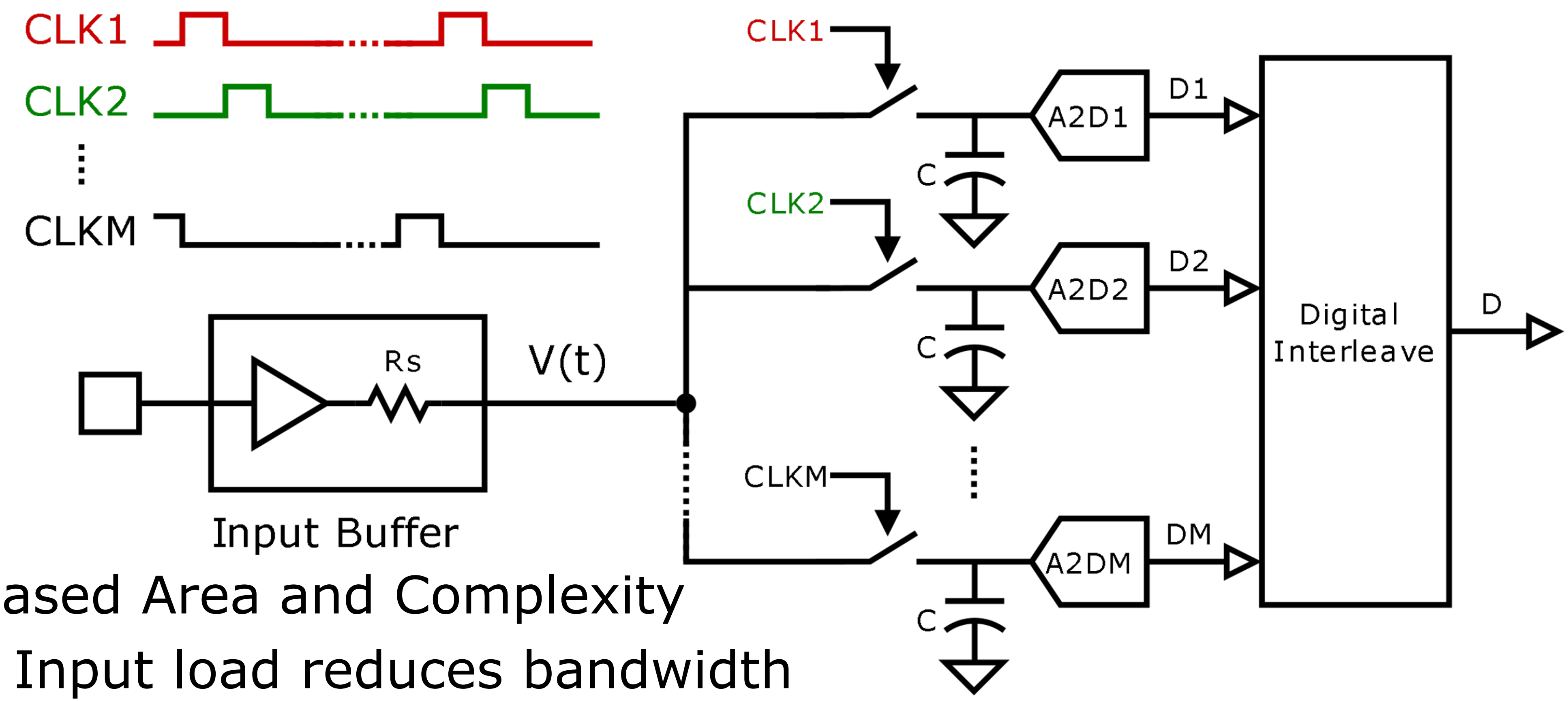
- □ Highest sample rate
- □ More power efficient at Fs > 1GS/s
  - ■ Highest Schreier Figure of Merit FOM_S*
- □ Interleaving M power efficient ADCs can be lower power than one fast ADC

ISSCC 1996-2019 [1]



*FOM_S=SNDR+10log10(P/BW)
P=Power, BW=Bandwidth

The Cost of Interleaving

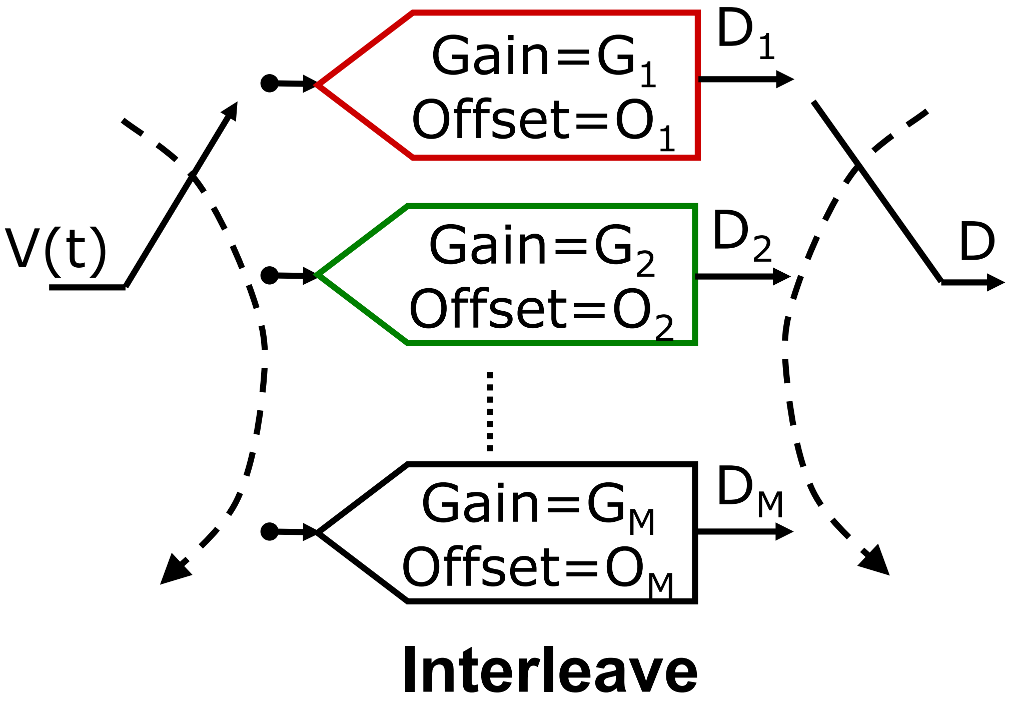


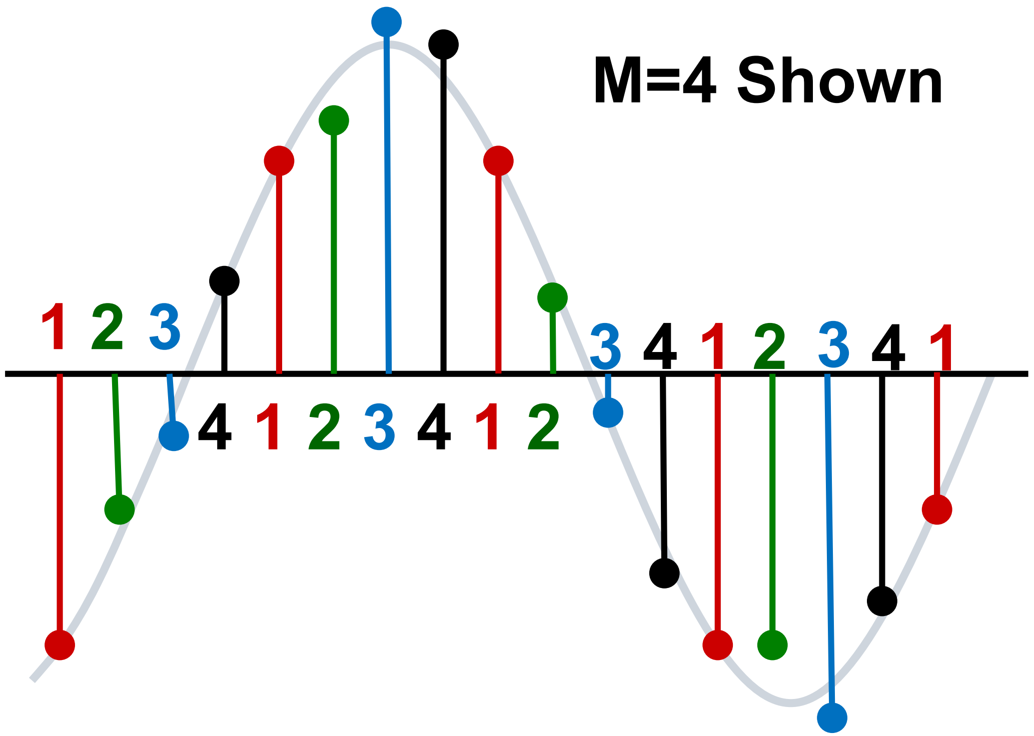
- □ Increased Area and Complexity
- □ More Input load reduces bandwidth
- □ New sources of error
  - Offset, Gain and timing Mismatch

Outline

- □ Background and Motivation
- □ Error Sources in Time-Interleaved ADCs
- □ Interim Q+A
- □ Frequency Spectrum of Interleave Errors
- □ Correction of Interleave Errors
- □ Time-Interleaved ADC Architectures.
- □ Summary and Conclusions

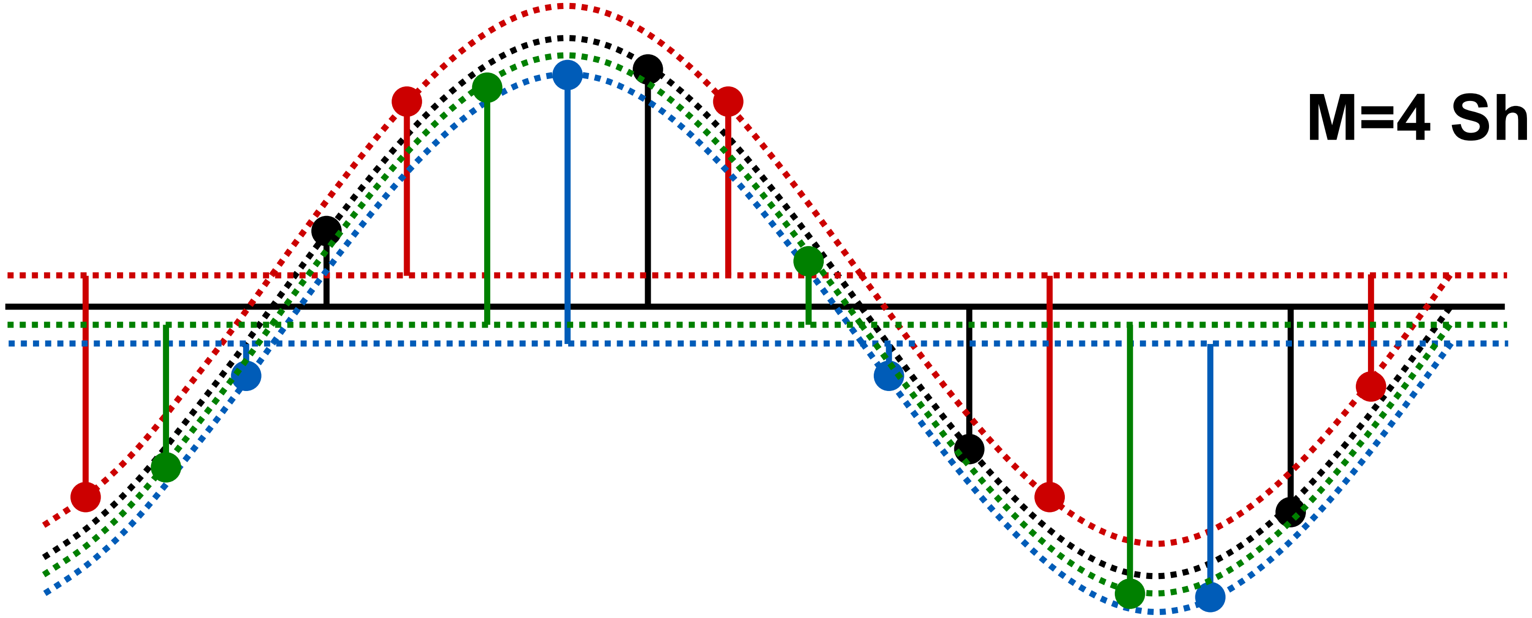
Time-Interleaved ADC Mismatch





- □ Equivalent to a single fast ADC if ADC Slices are identical
- □ But what if the ADC Slices don’t match?

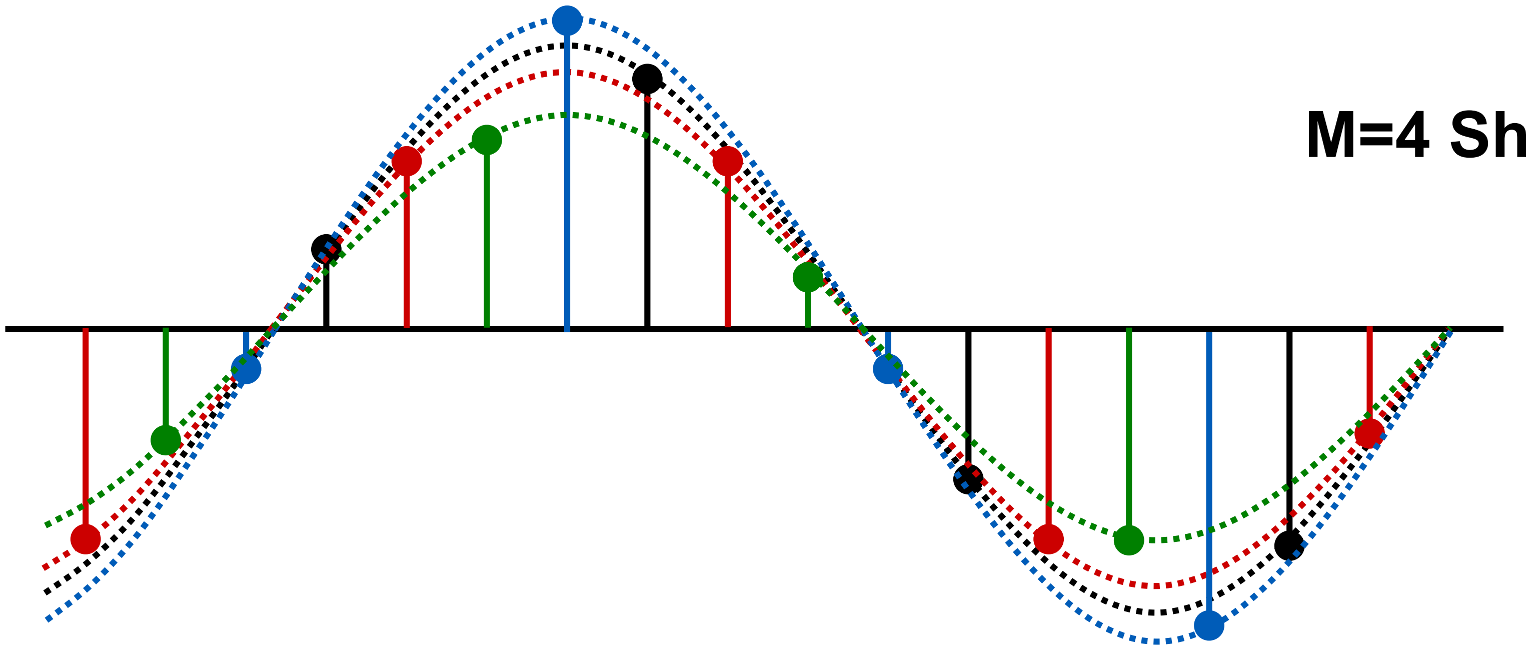
Offset Mismatch



M=4 Shown

- ☐ ADC slices have different offsets due to mismatch
- ☐ Looks like an added periodic offset with period M
- ☐ Always the same regardless of input amplitude or frequency

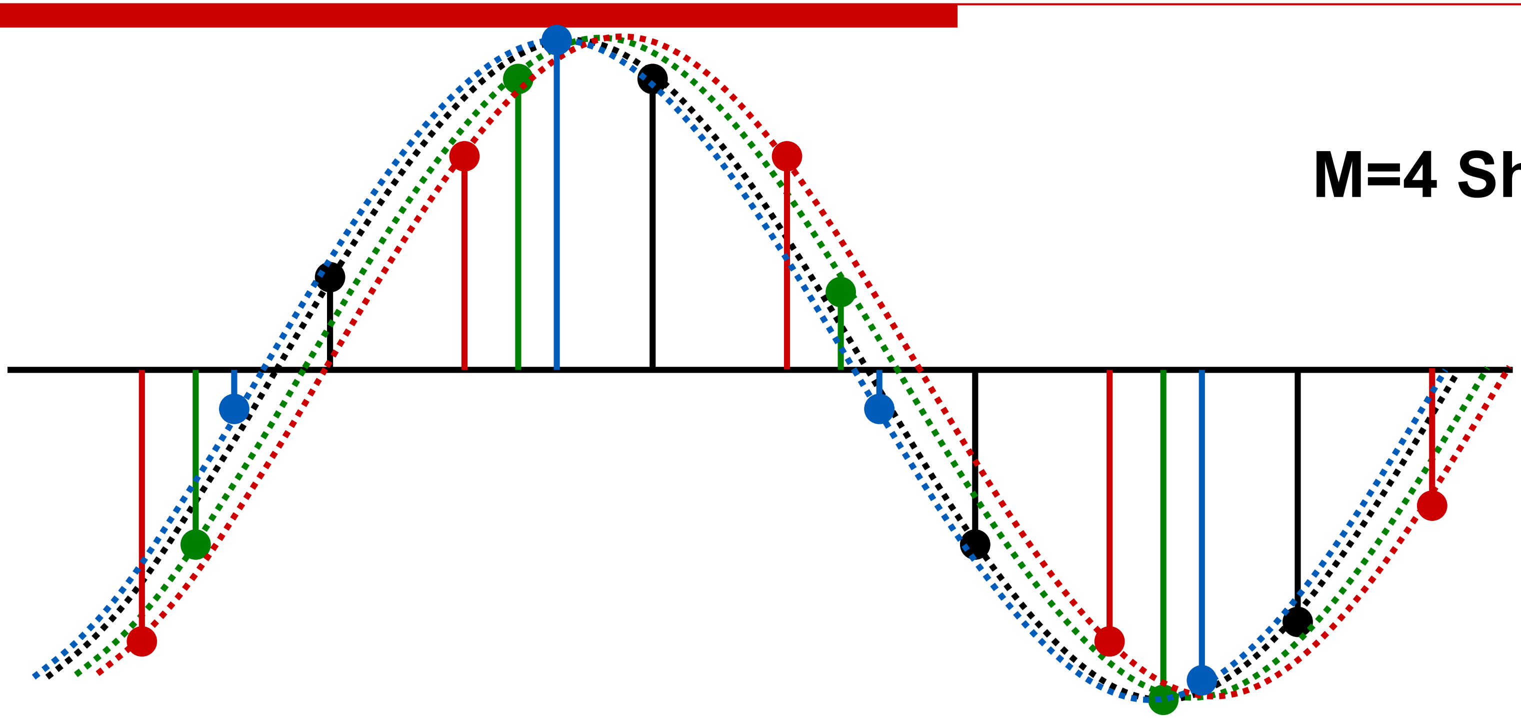
Gain Mismatch



M=4 Shown

- ☐ ADC Slices have different Gains due to mismatch
- ☐ Looks like amplitude is modulated by a waveform with period M
- ☐ Error is larger for large input signal amplitudes

Timing Mismatch



- ADC Slices do not sample T apart due to timing mismatch
- Looks like phase is modulated by a waveform with period M
- Error depends on signal slope when sampling
  - Larger error for large amplitudes and/or high frequencies

Outline

- □ Background and Motivation
- □ Error Sources in Time-Interleaved ADCs
- □ Interim Q+A
- □ Frequency Spectrum of Interleave Errors
- □ Correction of Interleave Errors
- □ Interleaved ADC Architectures.
- □ Summary and Conclusions

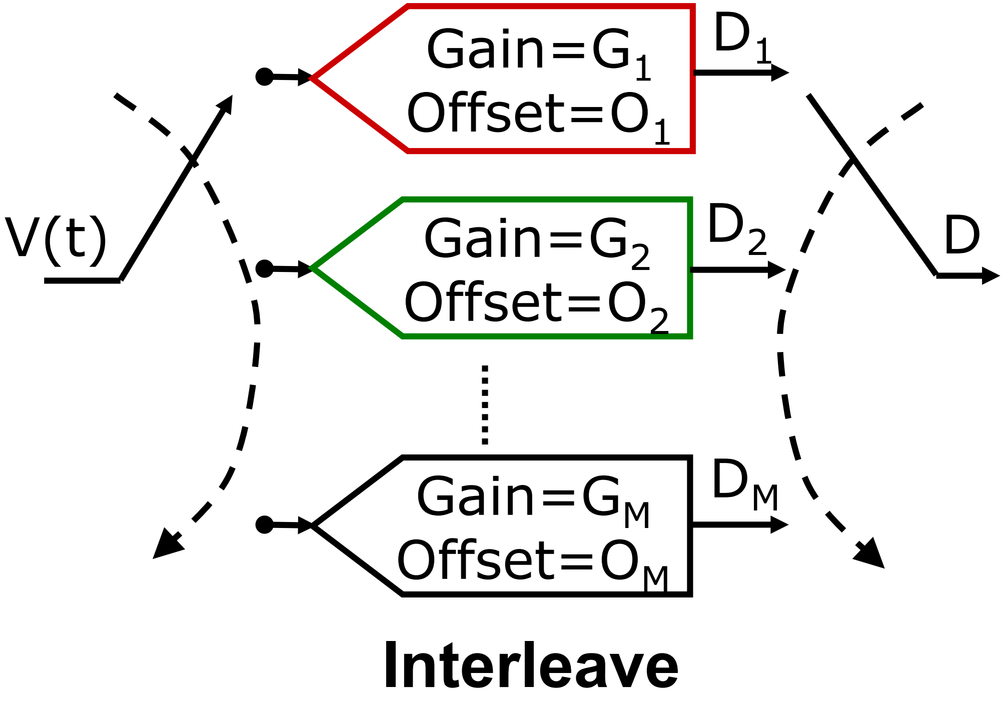
Interim Q+A Session (5 minutes)

Please ask questions that you feel are essential to follow the rest of this tutorial

Outline

- □ Background and Motivation
- □ Error Sources in Time-Interleaved ADCs
- □ Interim Q+A
- □ Frequency Spectrum of Interleave Errors
- □ Correction of Interleave Errors
- □ Time-Interleaved ADC Architectures.
- □ Summary and Conclusions

Frequency Spectrum of Interleave Errors



$$D_n[k]=G_nV(kT+t_n)+O_n$$

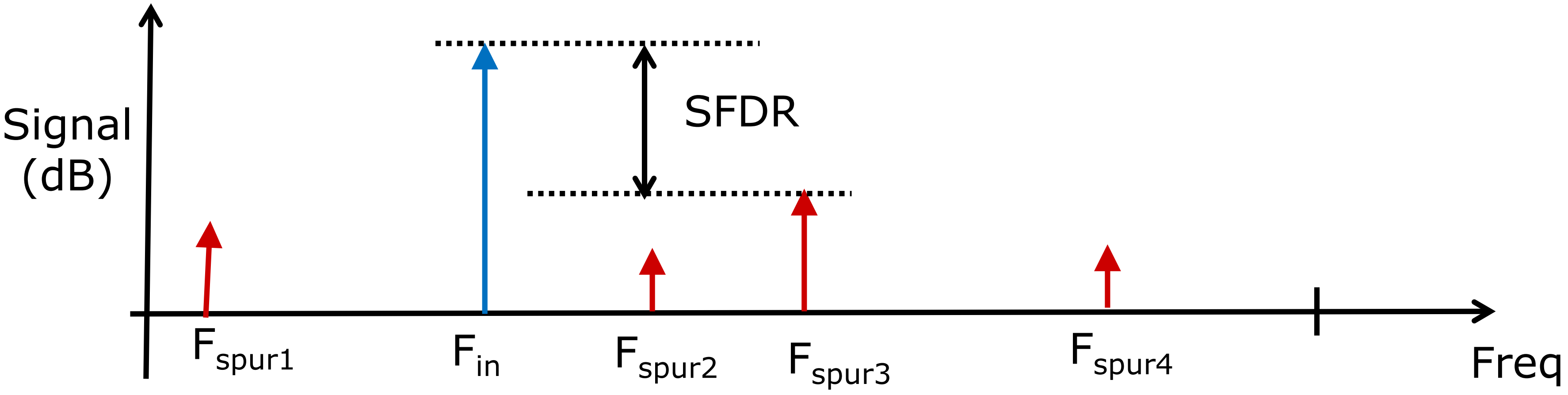
Gain Mismatch

Timing Mismatch

Offset Mismatch

- □ Output D[k] interleaved from M slice outputs $D_n[k]$
- □ $D_n[k]$ may mismatch in gain G_n , offset O_n and timing t_n
- □ How do these errors look in the frequency domain?

Frequency Spectrum of Interleave Errors



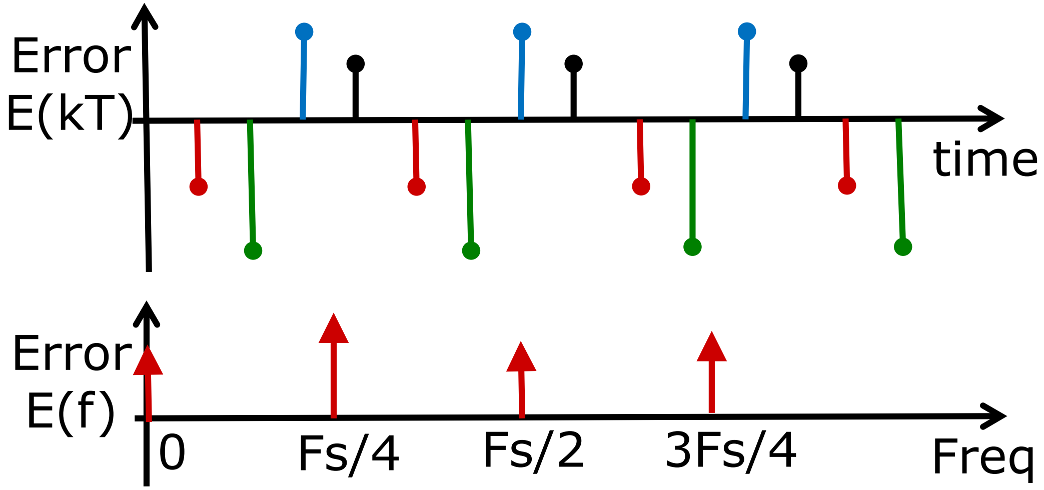
- ADC mismatch errors usually considered as noise
  - Limits Signal-to-Noise-and-Distortion Ratio (SNDR) of ADC
- But mismatch errors are concentrated at specific frequencies
  - Call these errors “spurs”
  - Limits Spurius-Free Dynamic Range (SFDR) of ADC
  - Important for Wireless applications

Offset Mismatch Errors

Sample	1	2	3	4	5
ADC 1	$G\cdot V(T)+O_1$				$G\cdot V(5T)+O_1$
ADC 2		$G\cdot V(2T)+O_2$			
ADC 3			$G\cdot V(3T)+O_3$		
ADC 4				$G\cdot V(4T)+O_4$	
D[k]	$G\cdot V(T)+O_1$	$G\cdot V(2T)+O_2$	$G\cdot V(3)+O_3$	$G\cdot V(4T)+O_4$	$G\cdot V(5T)+O_1$
E(kT)=error	O_1	O_2	O_3	O_4	O_1

- □ Assume all ADCs have the same gain G
- □ Each ADC has a different offset O_1,O_2,O_3, etc.
- □ Error E(kT) is sequence O_1,O_2,…,O_M,O_1,O_2,…
  - Repetitive sequence with period MT
- □ How does this error look in the frequency domain?

Frequency Spectrum of Offset Mismatch



M=4 Shown

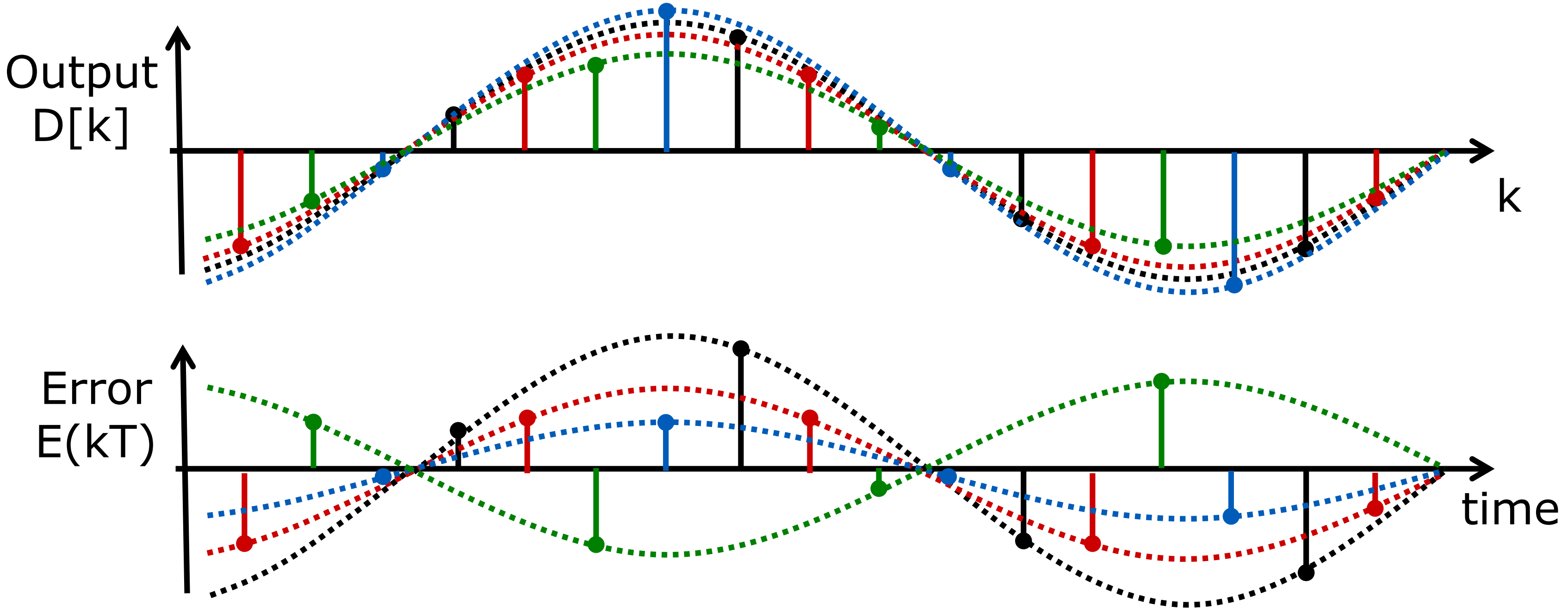
- Error is periodic sequence in time with period MT
- Fourier series representation
  - Tones at every integer multiples of F_S/M
  - Amplitude and phase of spurs depend on offset mismatch
  - Spurs are independent of input signal

Gain Mismatch Errors

Sample	1	2	3	4	5
ADC 1	$G_1 \cdot V(T)$				$G_1 \cdot V(5T)$
ADC 2		$G_2 \cdot V(2T)$			
ADC 3			$G_3 \cdot V(3T)$		
ADC 4				$G_4 \cdot V(4T)$	
D[k]	$G_1 \cdot V(T)$	$G_2 \cdot V(2T)$	$G_3 \cdot V(3T)$	$G_4 \cdot V(4T)$	$G_1 \cdot V(5T)$
E(kT)	$\Delta G_1 \cdot V(T)$	$\Delta G_2 \cdot V(2T)$	$\Delta G_3 \cdot V(3T)$	$\Delta G_4 \cdot V(4T)$	$\Delta G_1 \cdot V(5T)$

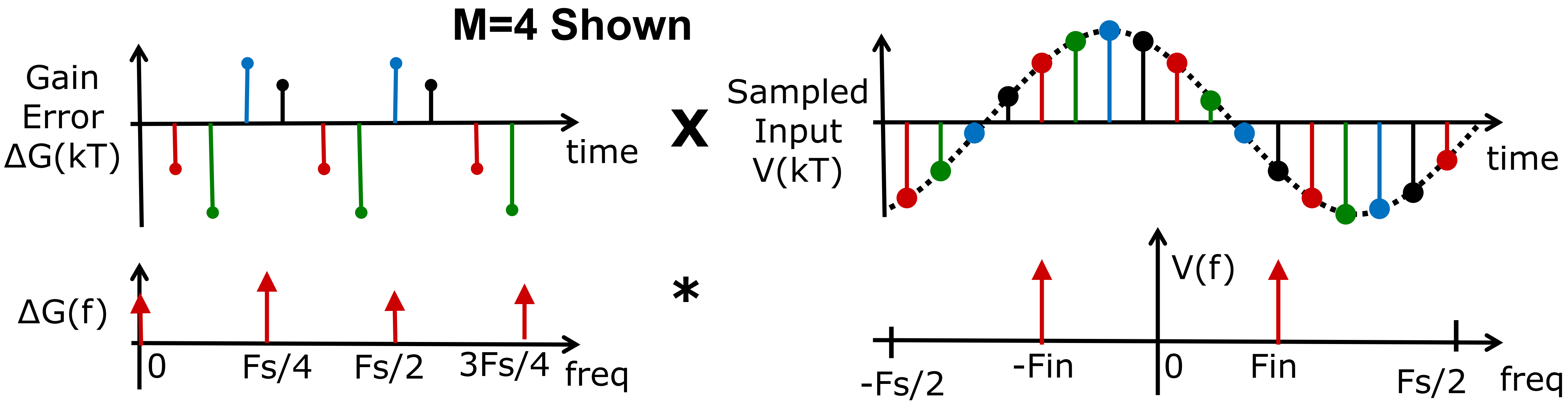
- Each ADC slice has a different gain G_1, G_2, G_3, etc.
  - Average gain is $G_{AVG}=(G_1+G_2+\ldots+G_M)/M$
- Slice n gain mismatch is $\Delta G_n=(G_n - G_{AVG})$
- Error E(kT) is ΔG_n multiplied by input sample V(kT)

Envelope of Gain Mismatch Errors



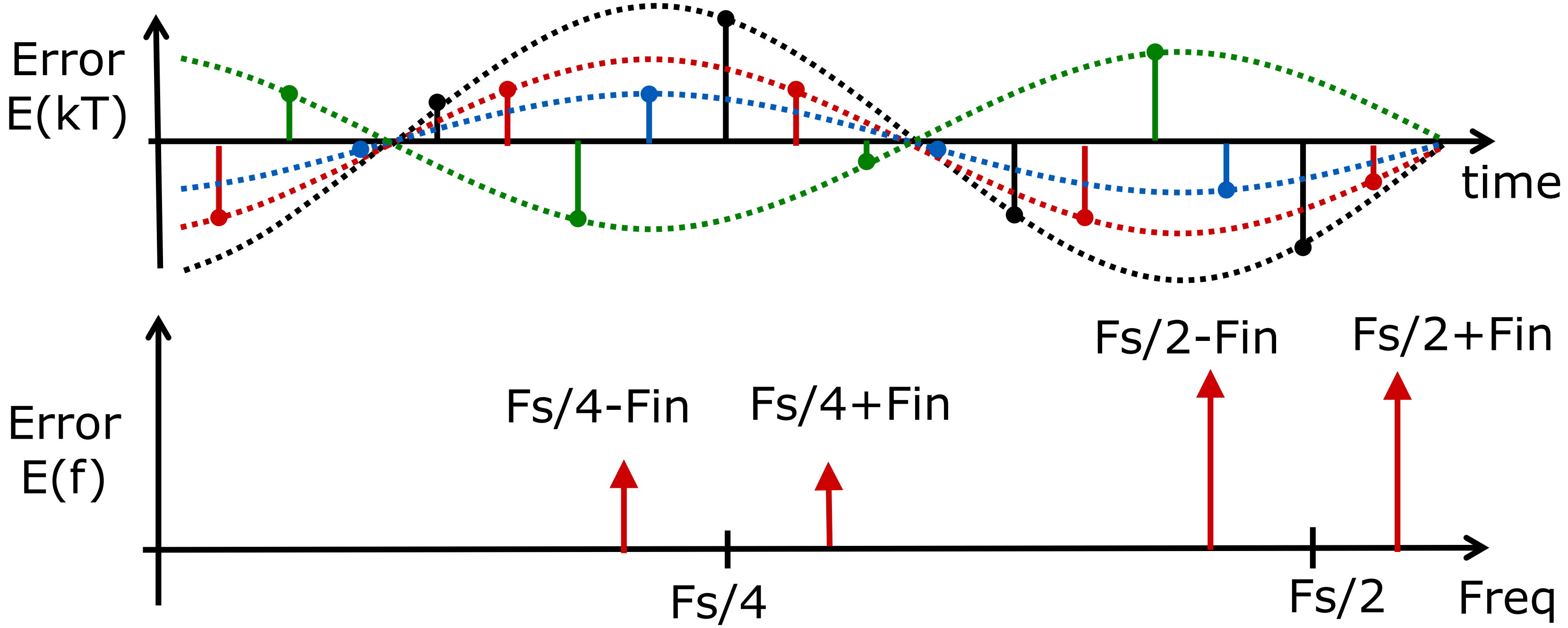
- □ Output D[k] consists of M interleaved signals
  - ■ Each has *envelope* of input V(t), but different Gain G_n
- □ Error E(kT) consists of M interleaved signals
  - ■ Each has *envelope* of input V(t), but different Gain $\Delta G_n=(G_n-G_{AVG})$

Frequency Spectrum of Gain Mismatch



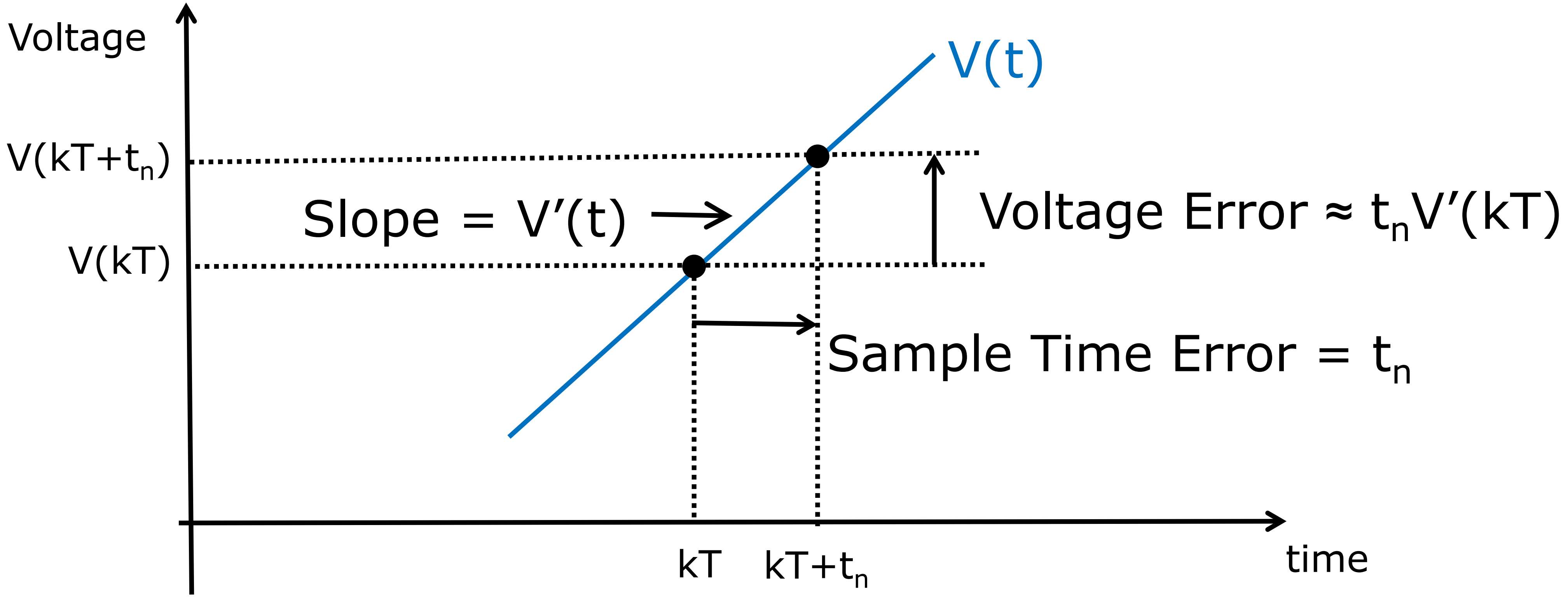
- Gain Error ΔG(kT) is periodic sequence in time with period MT
  - Fourier series representation ΔG(f) has tones at every integer $N\cdot F_S/M$
- Multiply ΔG(kT) by sampled input signal V(kT) with spectrum V(f)
  - Just like mixing or amplitude modulation
- Multiplication in time domain equivalent to *convolution* in frequency domain

Frequency Spectrum of Gain Mismatch



- E(f) has a scaled copy of input signal at every N·Fs/M±Fin
  - Amplitude and phase of spurs depends on gain mismatch
- Overall amplitude of E(f) increases with amplitude of input V(f)

Sample Time Error



□ Voltage error is slice timing error t_n multiplied by slope of V(t) :

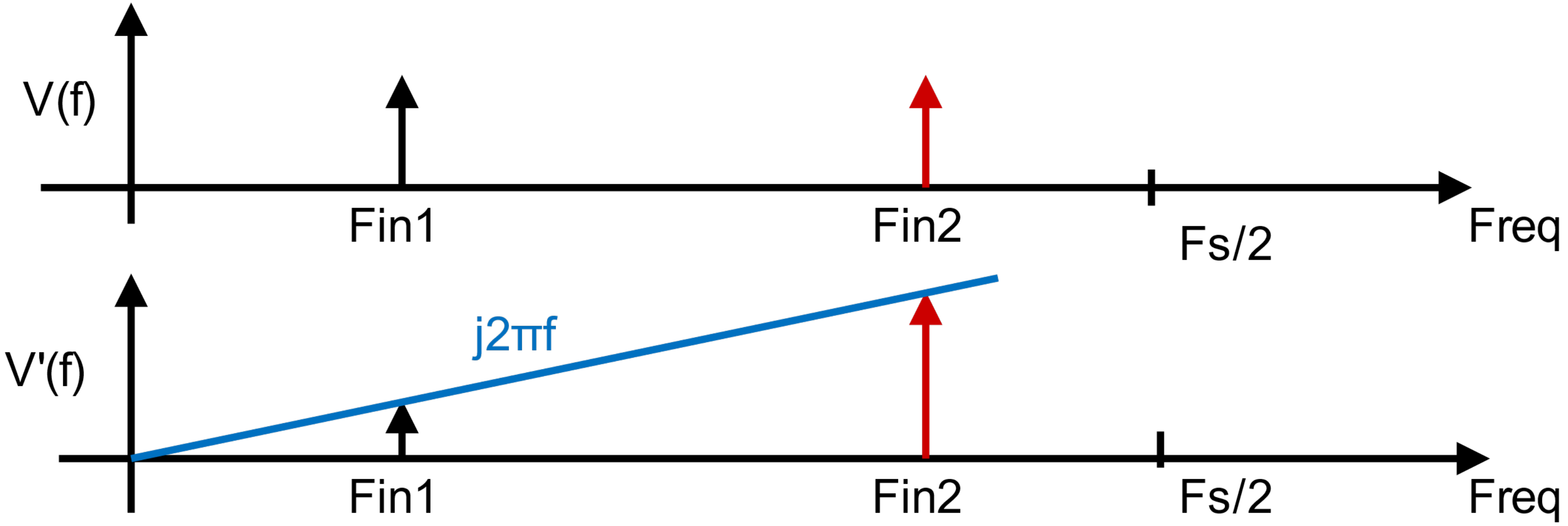
$$V'(kT) = \frac{dV}{dt} \ \ at\ t = kT$$

Timing Mismatch Errors

Sample	1	2	3	4	5
ADC 1	$G\cdot V(T+t_1)$				$G\cdot V(5T+t_1)$
ADC 2		$G\cdot V(2T+t_2)$			
ADC 3			$G\cdot V(3T+t_3)$		
ADC 4				$G\cdot V(4T+t_4)$	
D[k]	$G\cdot V(T+t_1)$	$G\cdot V(2T+t_2)$	$G\cdot V(3T+t_3)$	$G\cdot V(4T+t_4)$	$G\cdot V(5T+t_1)$
E(kT)	$G\cdot t_1\cdot V'(T)$	$G\cdot t_2\cdot V'(2T)$	$G\cdot t_3\cdot V'(3T)$	$G\cdot t_4\cdot V'(4T)$	$G\cdot t_1\cdot V'(5T)$

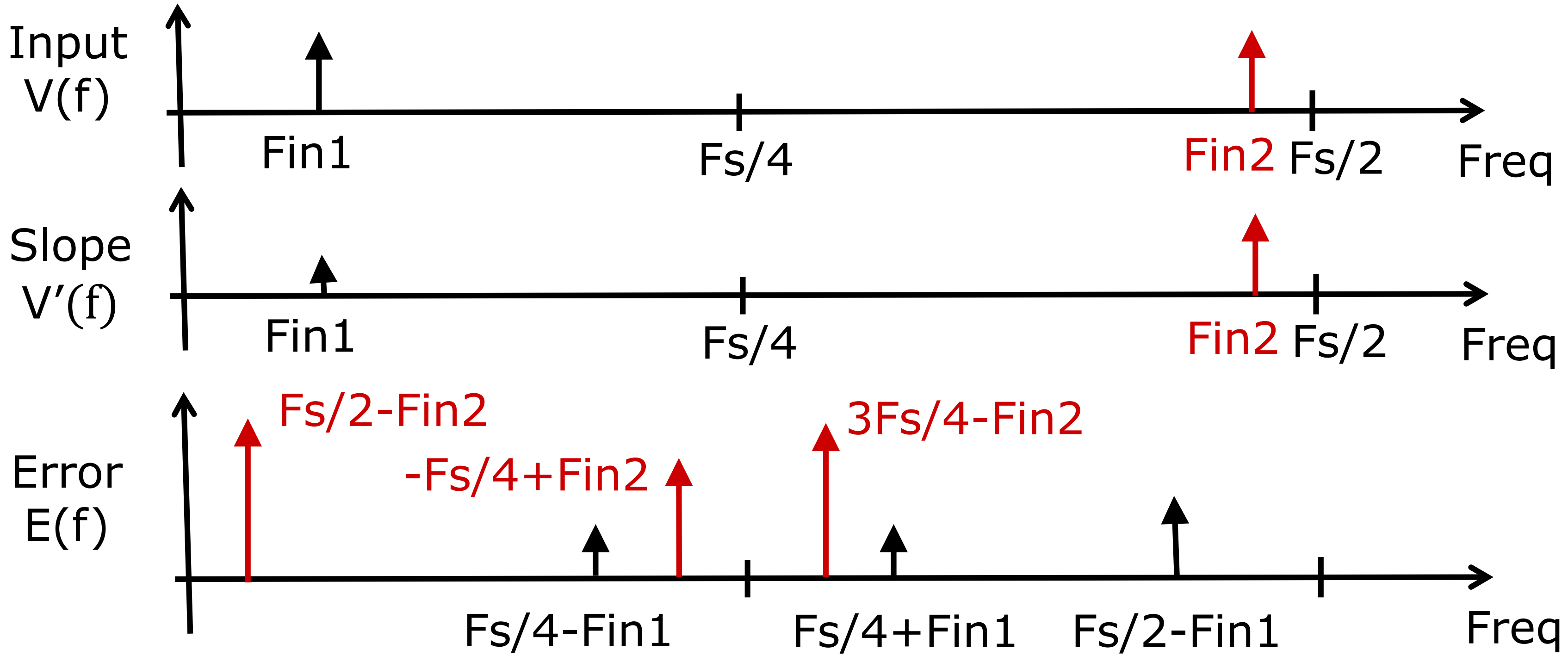
- Each ADC slice has a different timing error t_1,t_2,t_3, etc.
- Error E(kT) is slice timing error t_n multiplied by slope V'(t)
- Timing mismatch similar to gain mismatch
  - Multiplied by slope V'(t) instead of V(t)

Frequency Response of Slope Operation



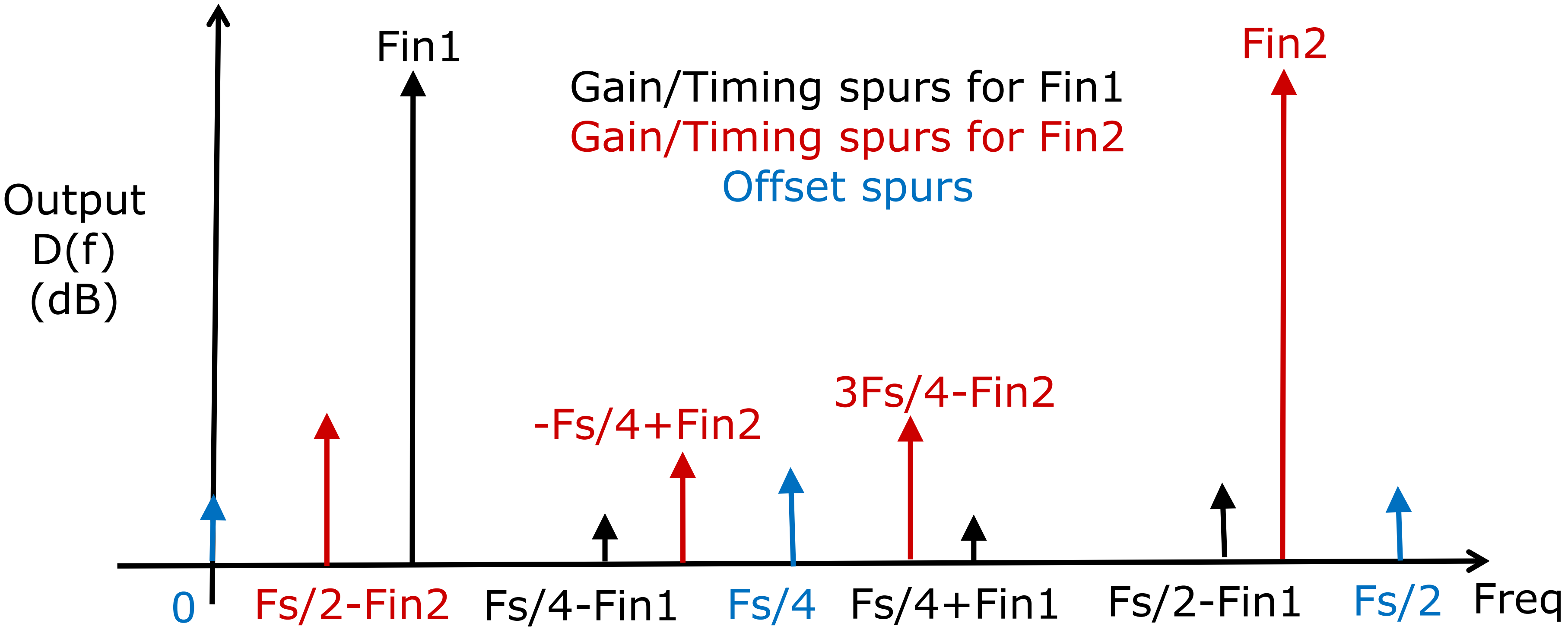
- Consider signal V(f) with two frequency components at Fin1 and Fin2
- Fourier transform : $V'(kT) = dV/dt \ \ => \ \ V'(f) = j2\pi f V(f)$
- High-pass response attenuates Fin1 relative to Fin2

Frequency Spectrum of Timing Mismatch



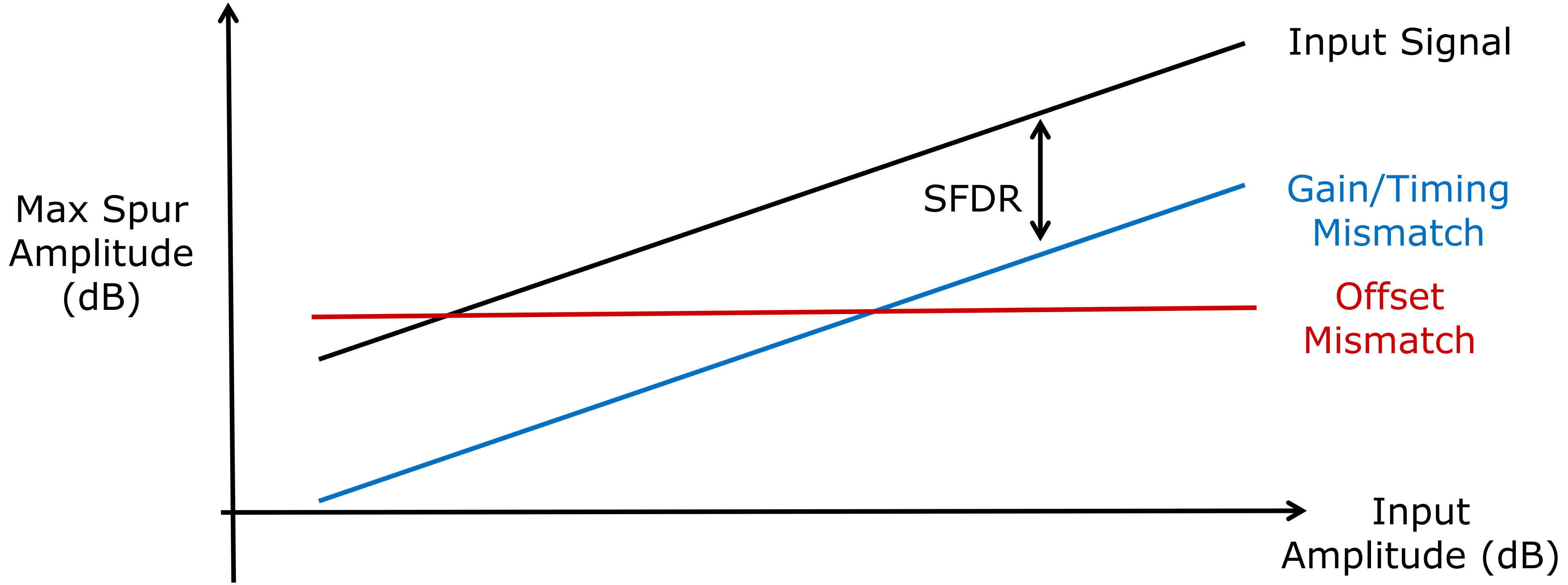
- □ Timing error like gain error with envelope of V′(t) instead of V(t)
  - No component at Fin1 or Fin2 as average timing mismatch is zero.
- □ E(f) has scaled alias of V′(f) at every N·Fs/M±Fin1 and N·Fs/M±Fin2
  - Amplitude of spurs increases with frequency and amplitude of input V(f)

ADC Output Spectrum with M=4



- □ Overall spectrum of ADC output code D(kT) is D(f)
  - ■ Offset spurs at N·Fs/M => independent of input signal
  - ■ Gain/Timing spurs at N·Fs/M ± Fin

SFDR vs. Input Amplitude



- Offset mismatch limits SFDR at small input amplitude
- Gain and Timing mismatch limits SFDR at large input amplitude
  - Mostly gain mismatch at low frequencies, timing mismatch at high frequencies

Summary So Far

- Time-interleaved ADCs popular for >1GS/s
  - Highest sample rate
  - More power efficient
- Mismatch spurs limit performance
  - Offset spurs at fixed amplitude and frequency
  - Gain/Timing spurs at alias frequencies of interleave rate
- How do we implement a time-interleaved ADC to minimize these effects?

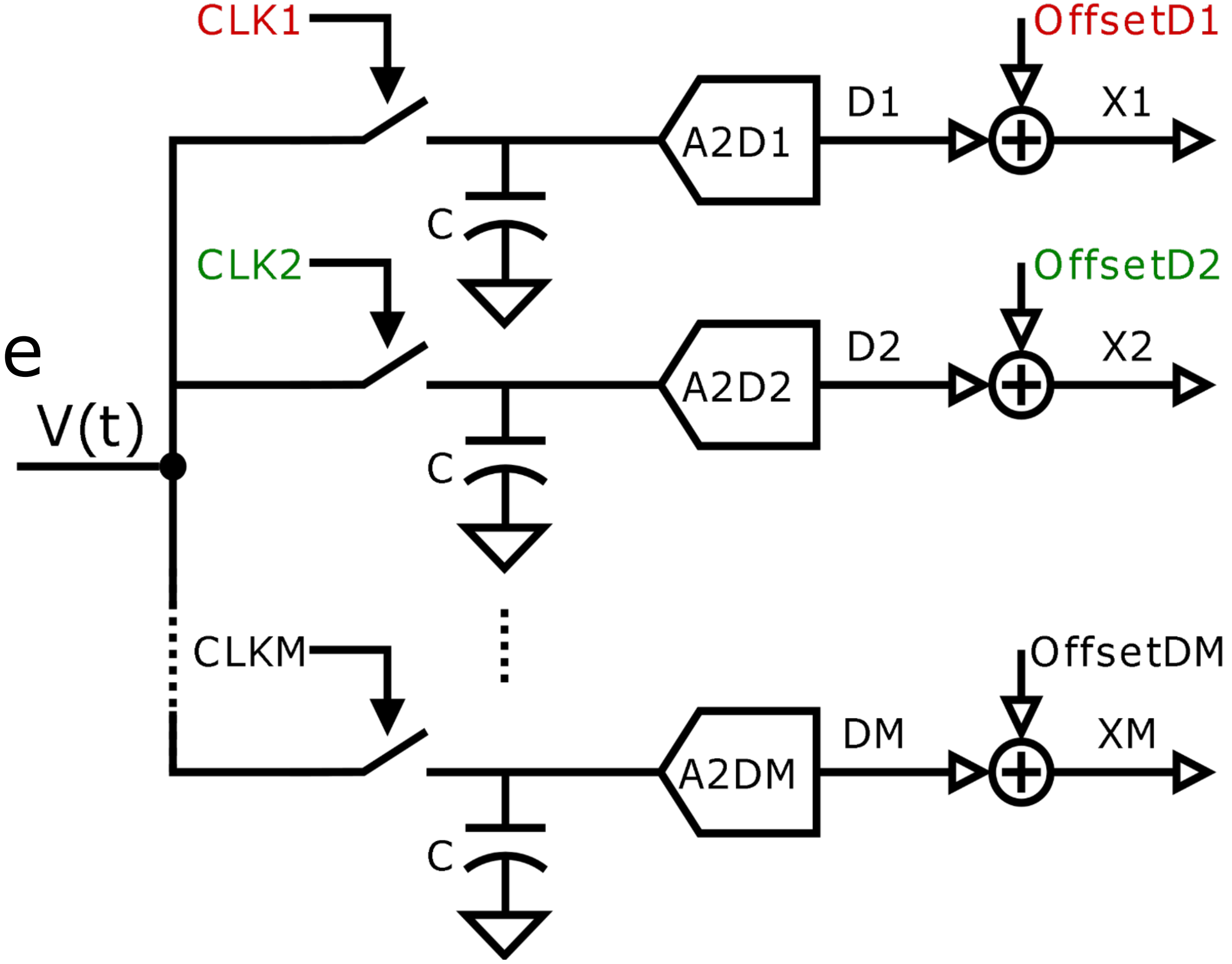
Outline

- □ Background and Motivation
- □ Error sources in Time-Interleaved ADCs
- □ Interim Q+A
- □ Frequency Spectrum of Interleave Errors
- □ Correction of Interleave Errors
- □ Time-Interleaved ADC Architectures.
- □ Summary and Conclusions

Correcting Offset Mismatch - Digital

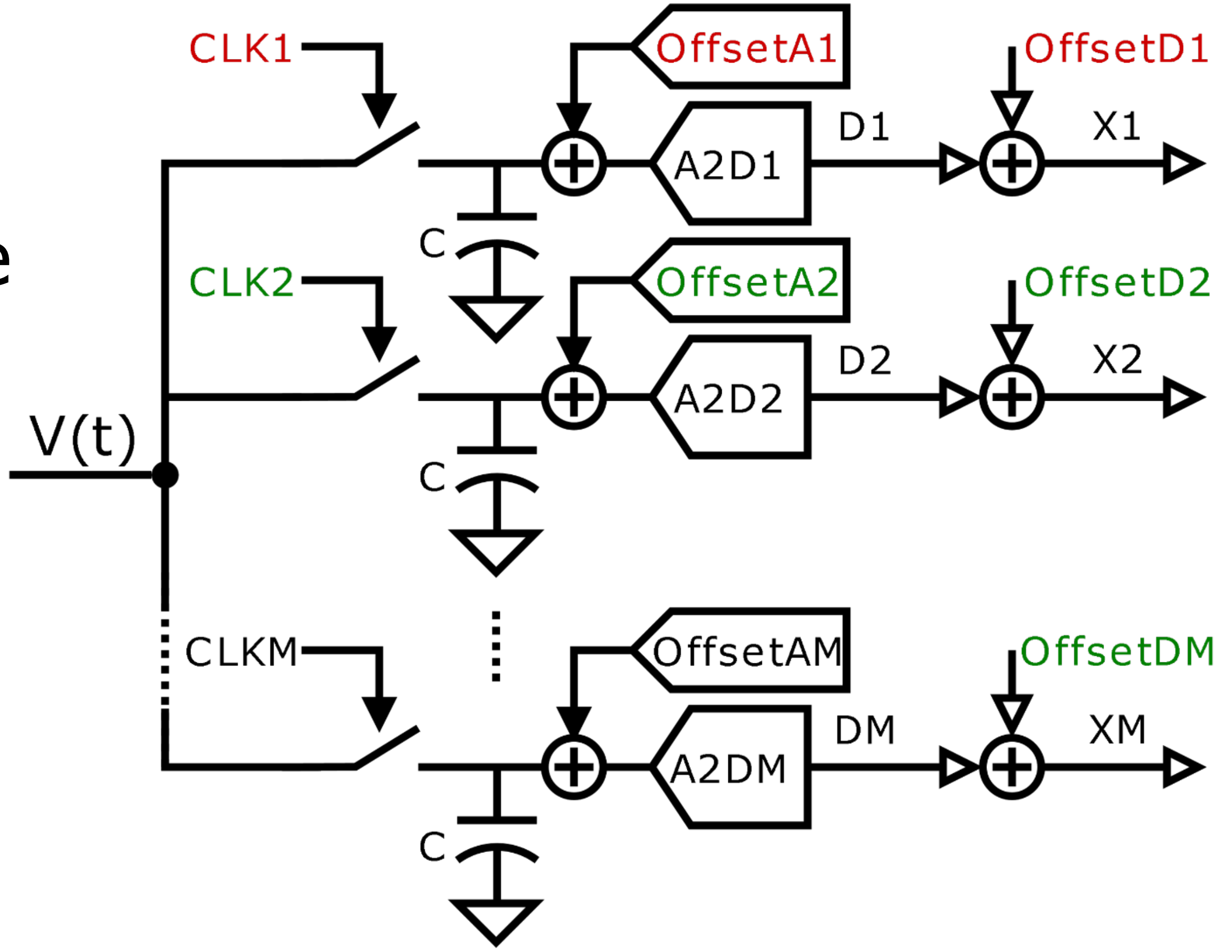
- □ Digital Correction
  - Add offset digitally
- ✓ Unlimited precision
- × Reduces full scale range
  - ADC Clipping occurs before correction

$$\mathrm{OffsetDn} = -G_nO_n$$



Correcting Offset Mismatch - Analog

- □ Analog Correction
  - ■ Inject offset using DAC
- ✓ Maintains full scale range
- × Limited precision
- □ Usually combine both Analog and Digital
  - ■ Analog Coarse
  - ■ Digital Fine

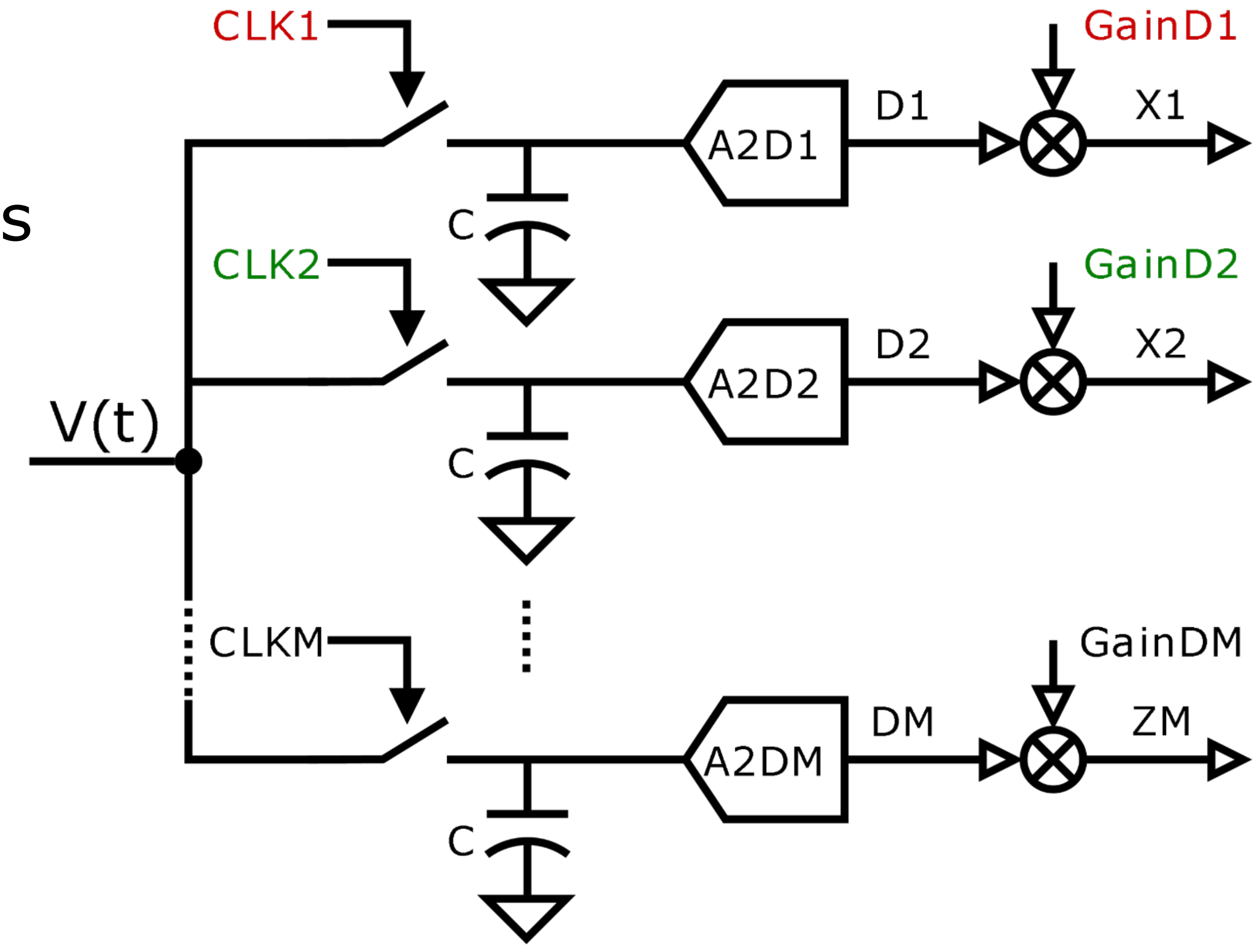


$$\text{OffsetAn}+\text{OffsetDn}/G_n=-O_n$$

Correcting Gain Mismatch - Digital

- □ Digital Correction
  - ■ Use Digital Multipliers
- ✓ Unlimited precision
- × Multipliers can be expensive

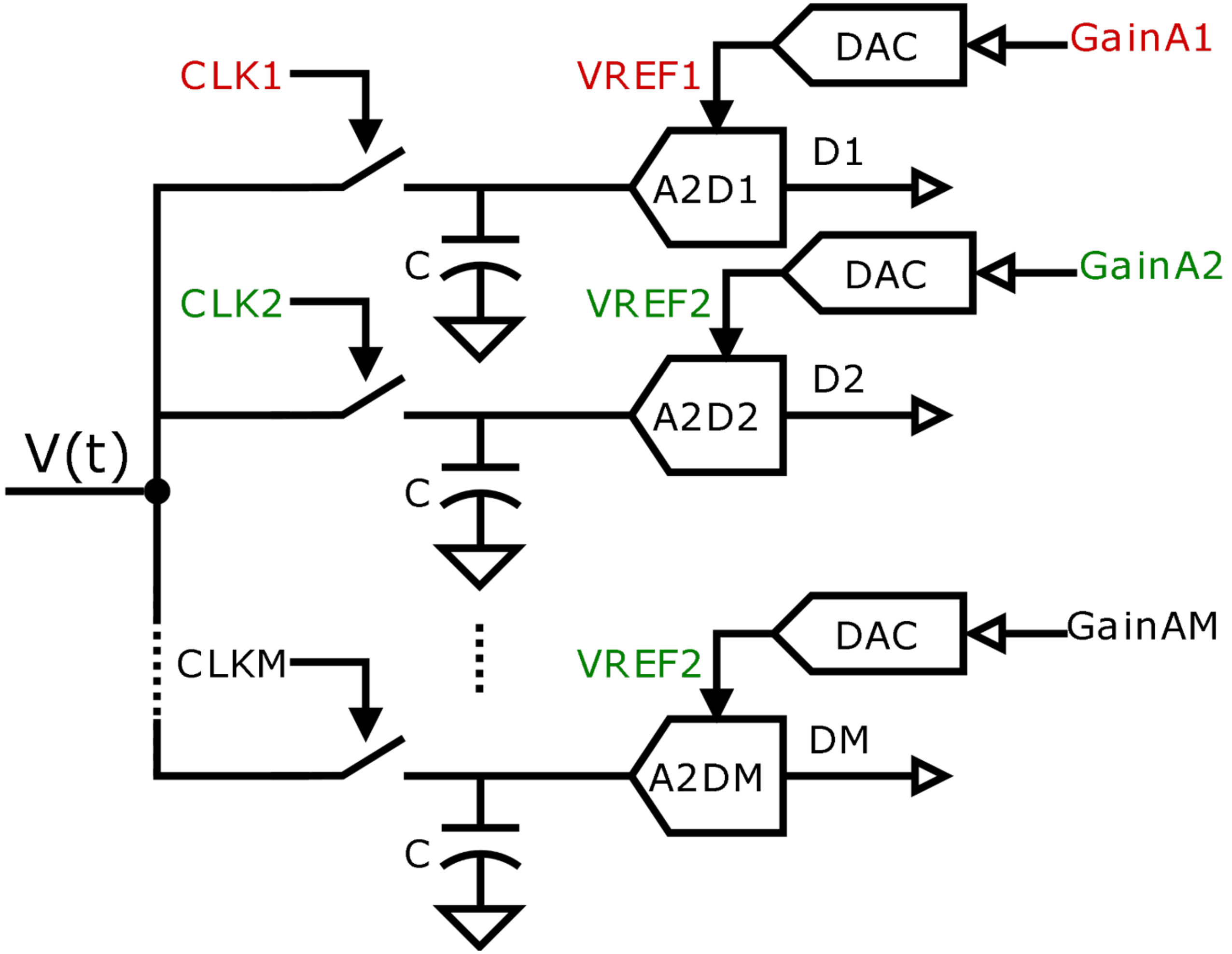
$$\text{GainDn} = G_{AVG}/G_n$$



Correcting Gain Mismatch - Analog

- □ Analog Correction
  - Use DACs to adjust VREF per slice
- ✓ Low Power
- × Limited precision
  - Use for low resolution applications

GainAn = G_{AVG}/G_n



Correcting Timing Mismatch - Analog

□ Analog Correction

- Adjustable Delay CLK Buffer

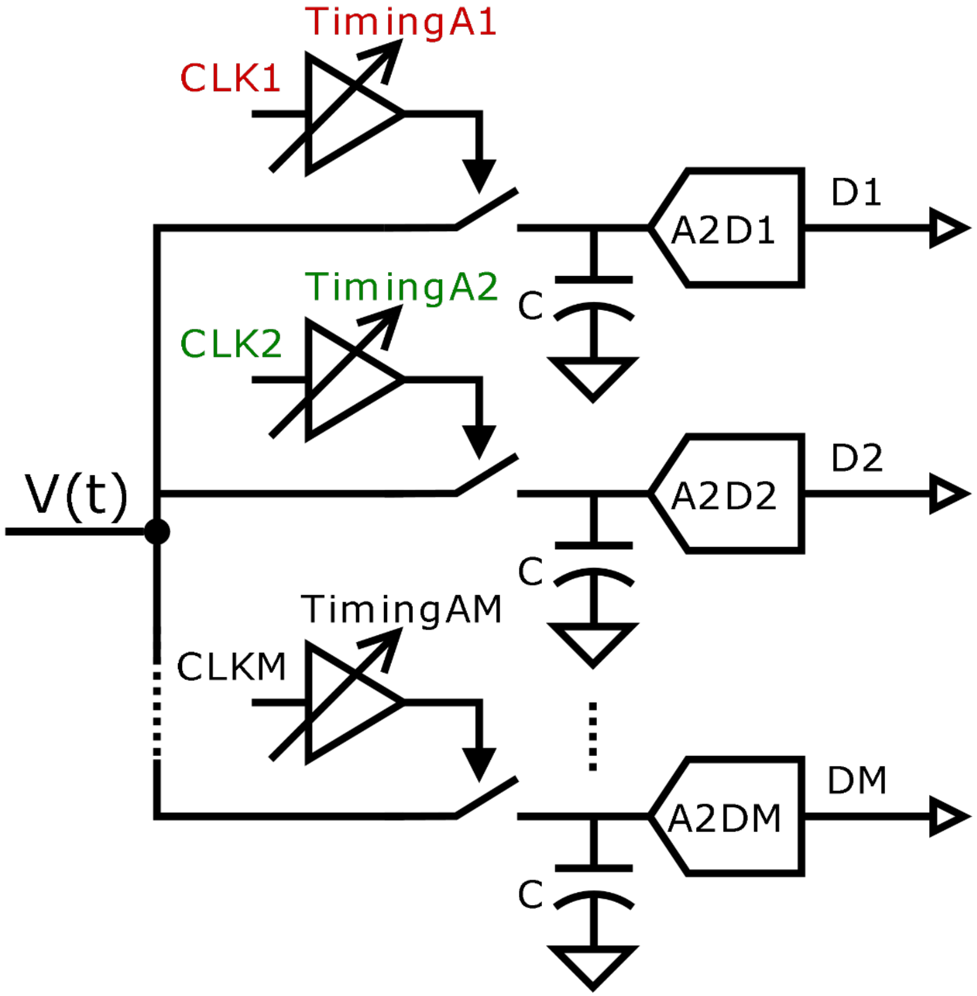
✓ Low cost implementation

✓ Can be very precise

× Extra delay in sampling clock

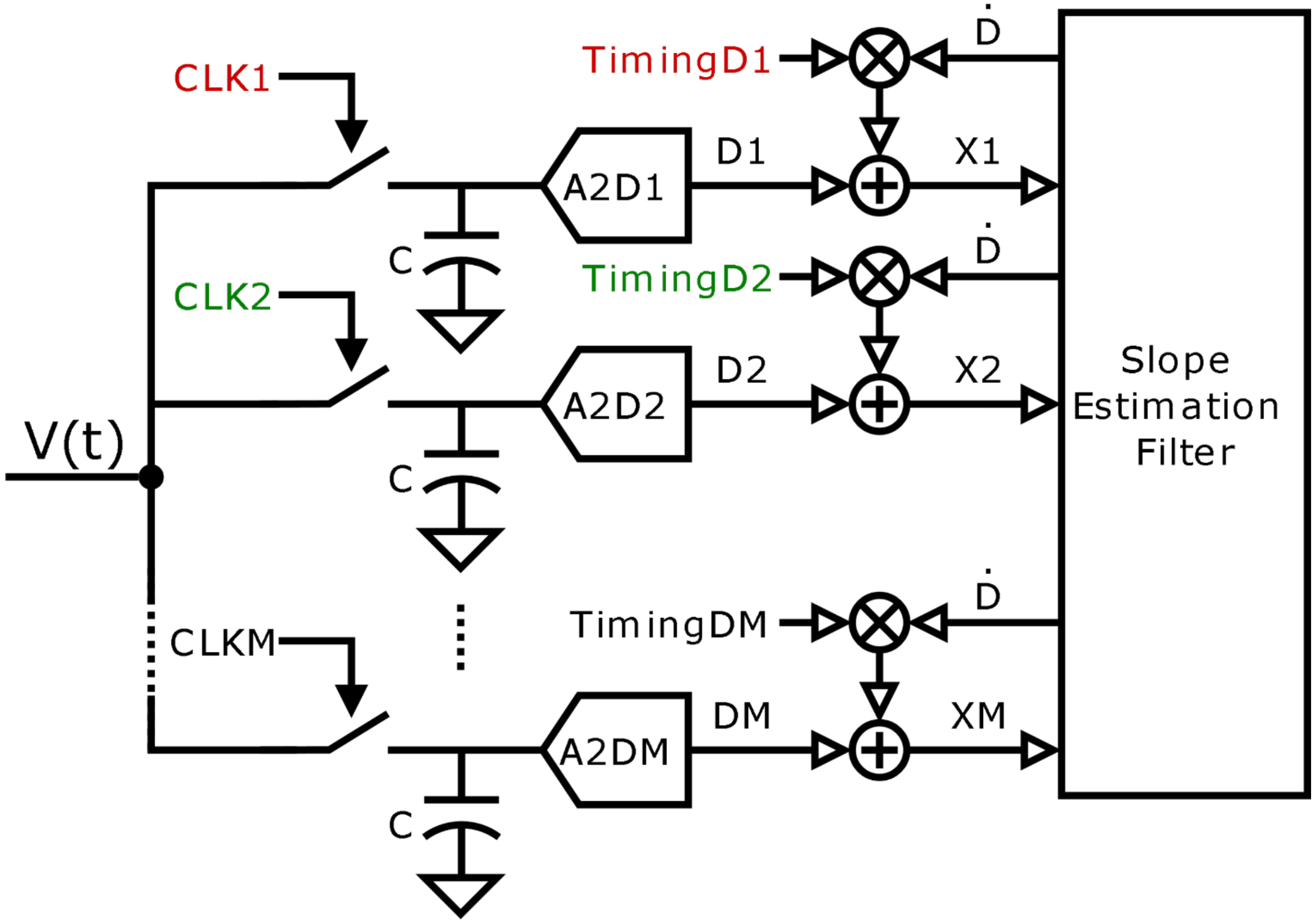
- Increases random delay variation (jitter)

$$\text{TimingAn} = -t_n$$



Correcting Timing Mismatch - Digital

- □ Digital Correction
  - ■ Estimate Slope $\dot{D}$
- ✓ Simple clock path
  - ■ Lowest jitter
- × Expensive DSP
- × Only works in one Nyquist band
- × Limited to ~90% of Nyquist frequency



$$\text{TimingDn} = -t_n$$

Measuring Mismatch Errors

- How do we measure mismatch errors so they can be corrected?
  - Complex topic with a lot of active research (See Ref. [c])

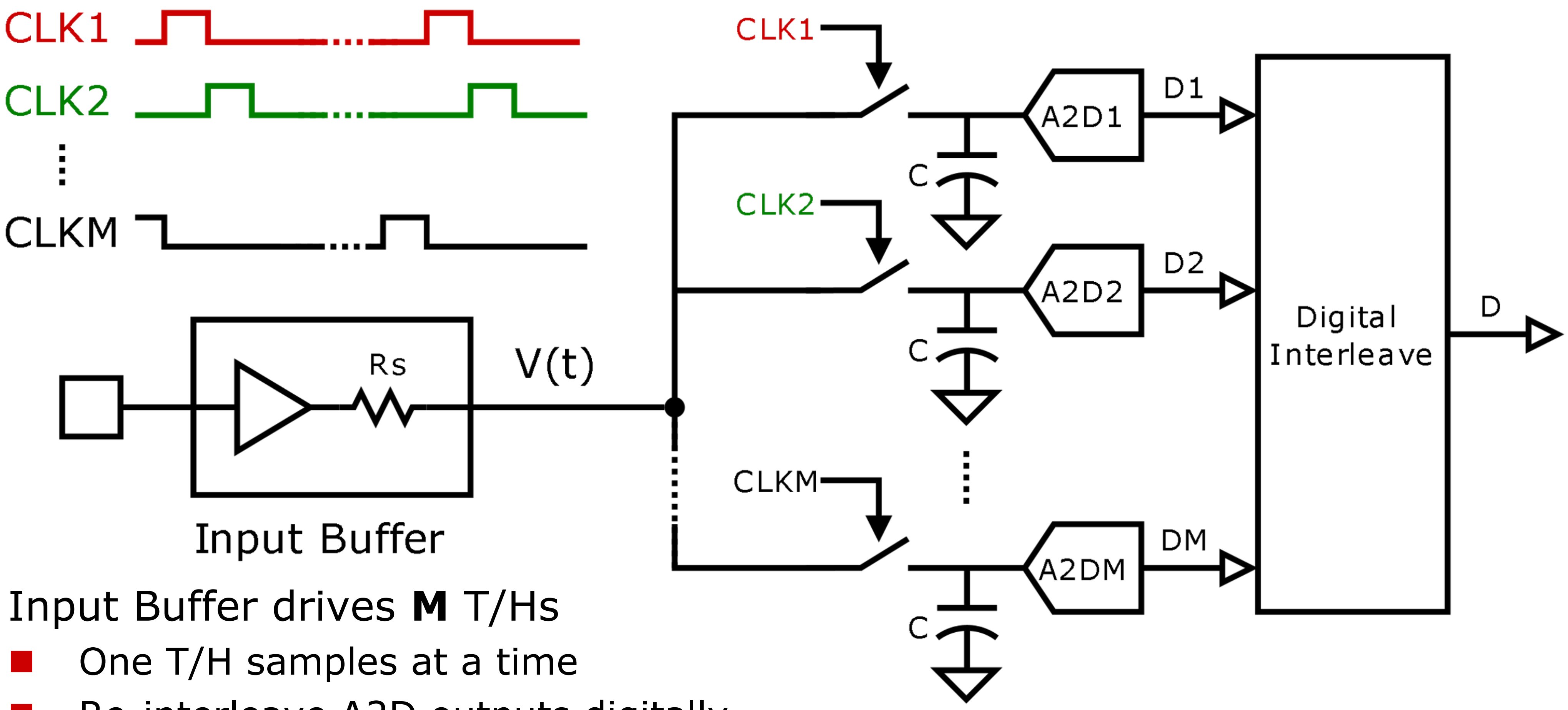
- Foreground/Offline
  - Apply known input signal V(t) e.g. sinewave of known frequency.
  - Analyze output of ADC Slices for differences in offset, gain, phase.
  - Must interrupt operation to re-calibrate for temperature, supply drift etc.

- Background/Online
  - Works while ADC is converting normal signal
  - Allows tracking temperature, supply changes etc.
  - Many methods analyze statistics of ADC outputs
    - Not always robust for all input signals
    - What if the input signal looks like a mismatch error?

Outline

- □ Background and Motivation
- □ Error Sources in Time-Interleaved ADCs
- □ Frequency Spectrum of Interleave Errors
- □ Interim Q+A
- □ Correction of Interleave Errors
- □ Time-Interleaved ADC Architectures
- □ Summary and Conclusions

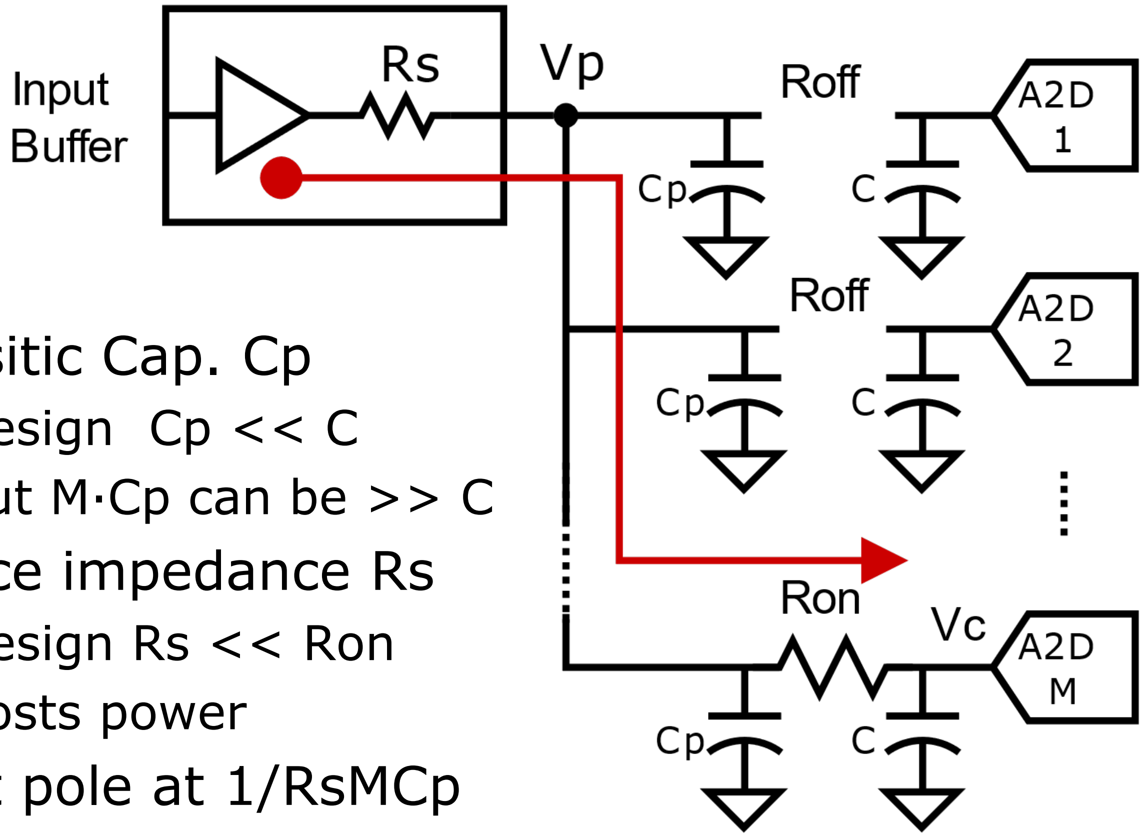
Time-Interleaved ADC Architecture

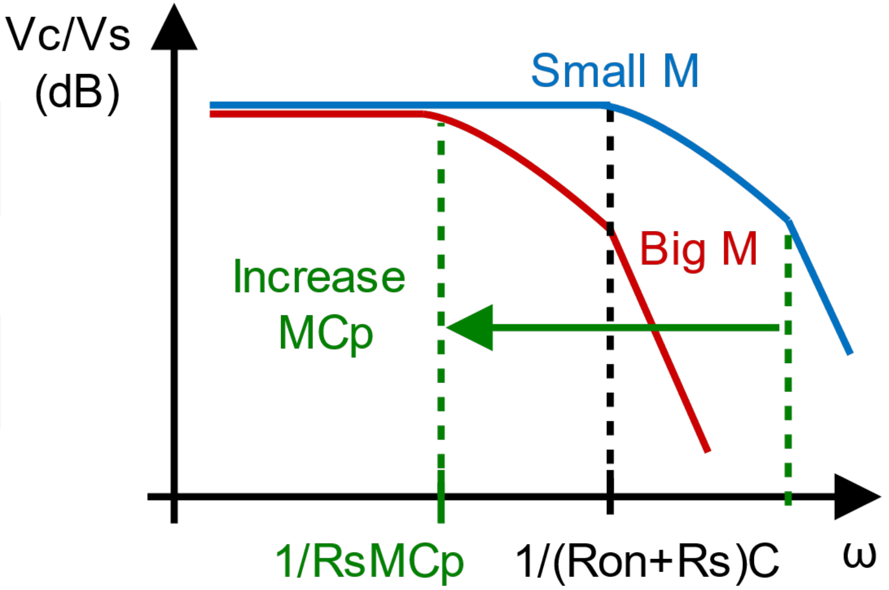


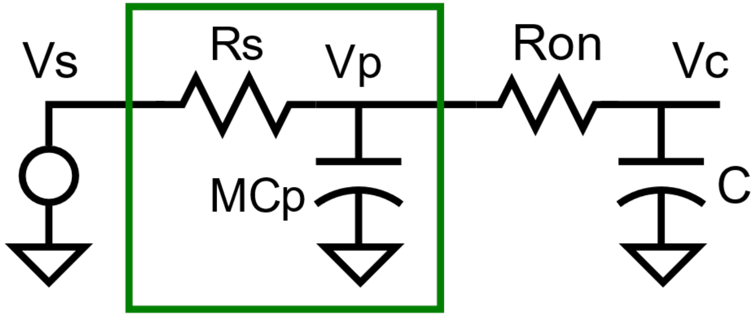
- Input Buffer drives **M** T/Hs
  - One T/H samples at a time
  - Re-interleave A2D outputs digitally

Sampling Bandwidth

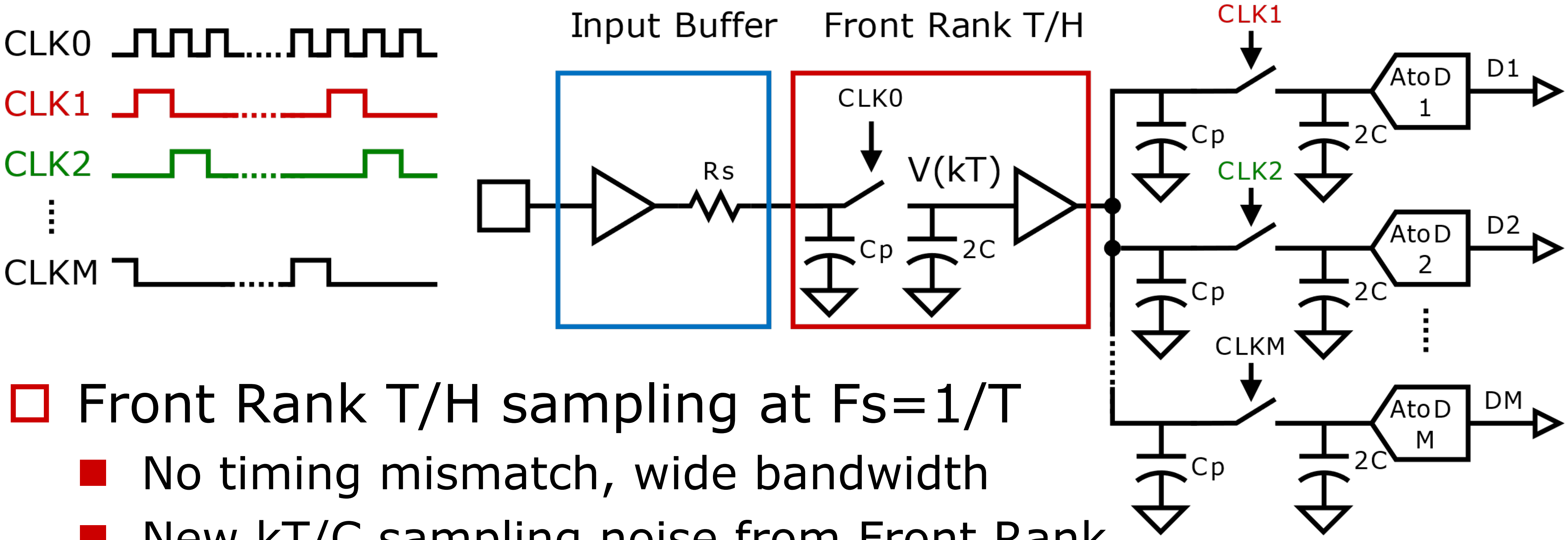
- ☐ Parasitic Cap. Cp
  - ■ Design Cp << C
  - ■ But M·Cp can be >> C
- ☐ Source impedance Rs
  - ■ Design Rs << Ron
  - ■ Costs power
- ☐ Input pole at 1/RsMCp
  - ■ Can limit Bandwidth!





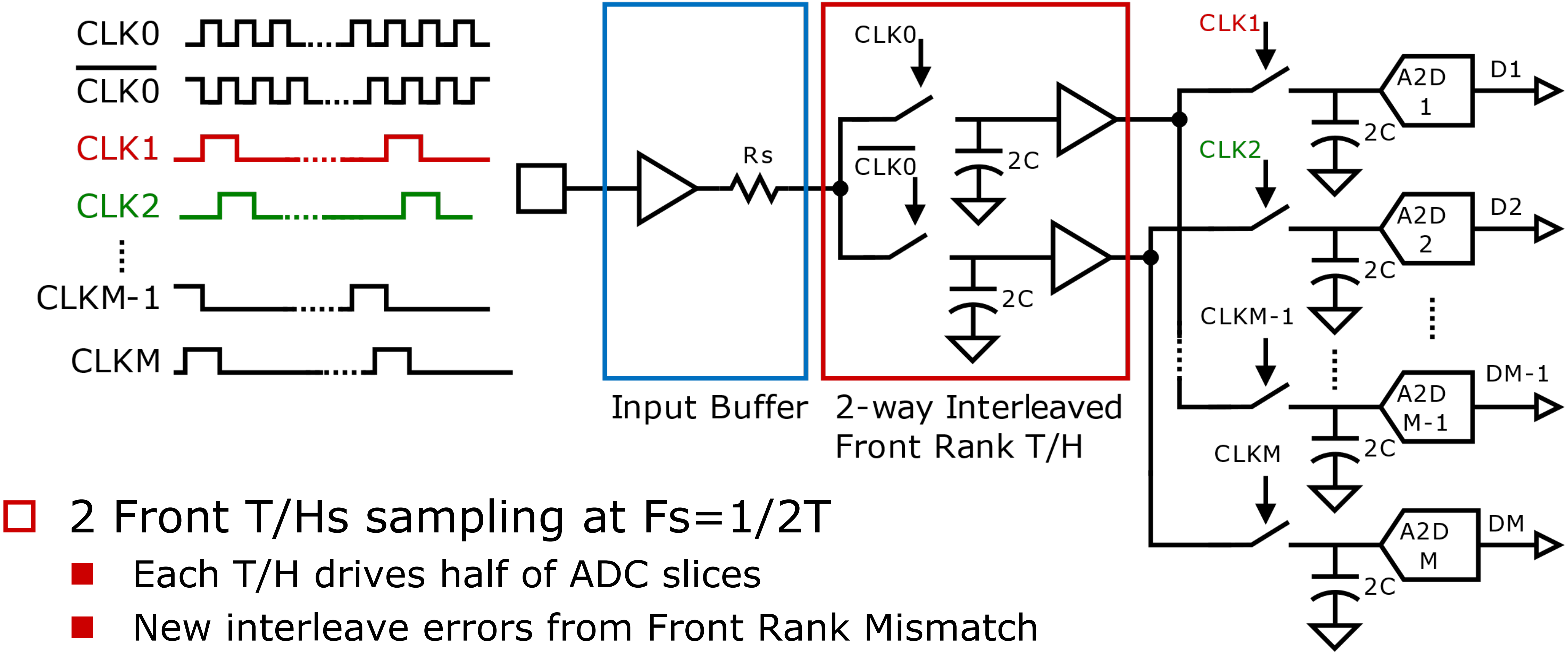


Front Rank T/H [3]



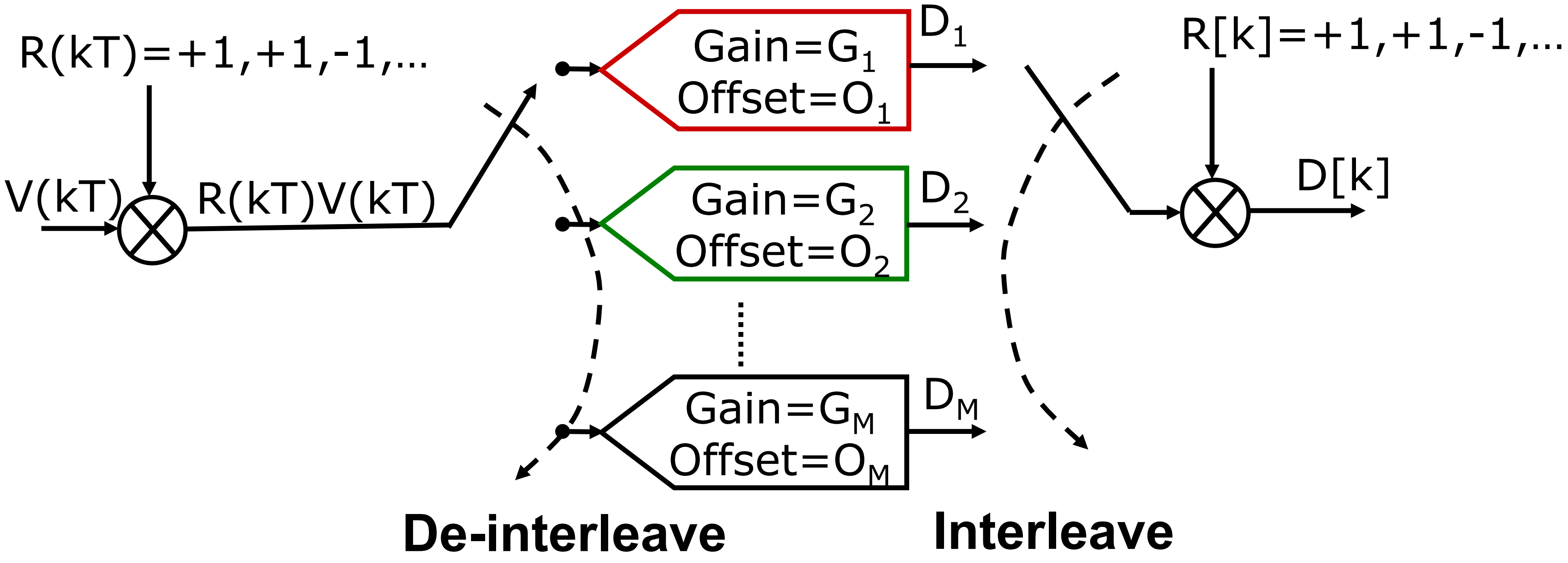
- □ Front Rank T/H sampling at Fs=1/T
  - No timing mismatch, wide bandwidth
  - New kT/C sampling noise from Front Rank
    - □ E.g. ADC slice C => 2C to keep total kT/C noise the same
  - T/H still drives M slices => Lots of extra power!

2-way Interleaved Front Rank T/H



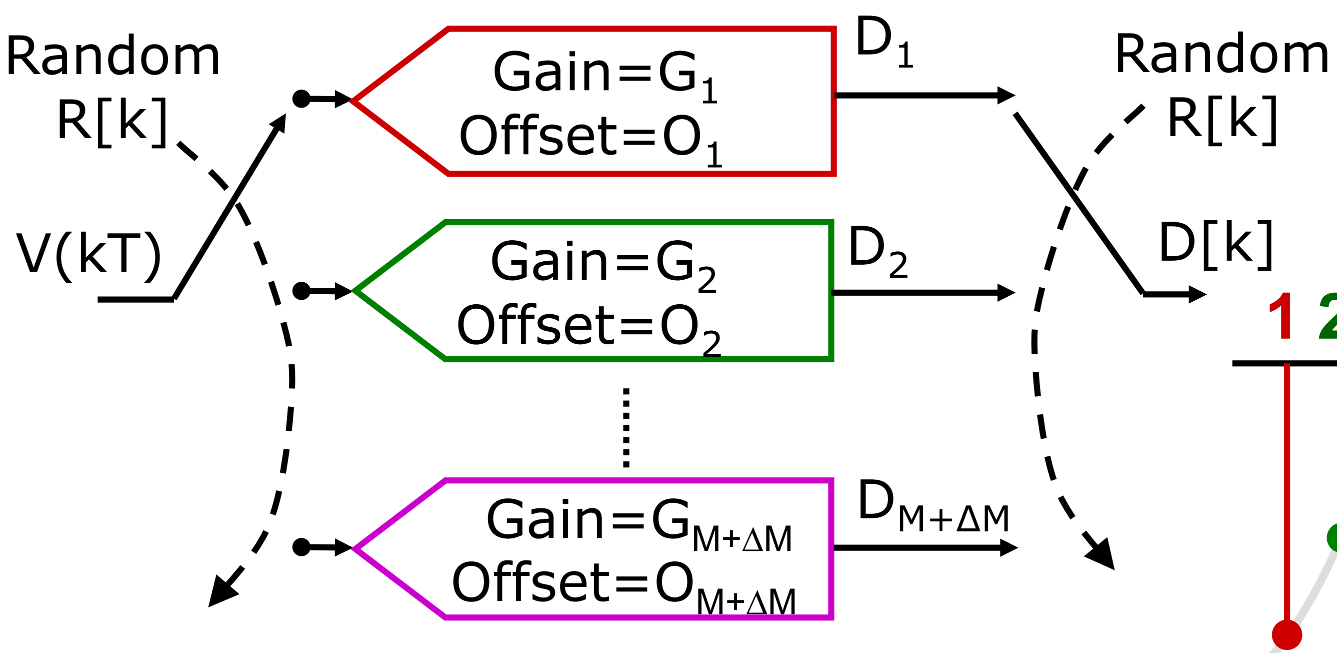
- □ 2 Front T/Hs sampling at Fs=1/2T
  - Each T/H drives half of ADC slices
  - New interleave errors from Front Rank Mismatch
  - Can extend Front Rank interleaving to 4 or more

Random Chopping [5]

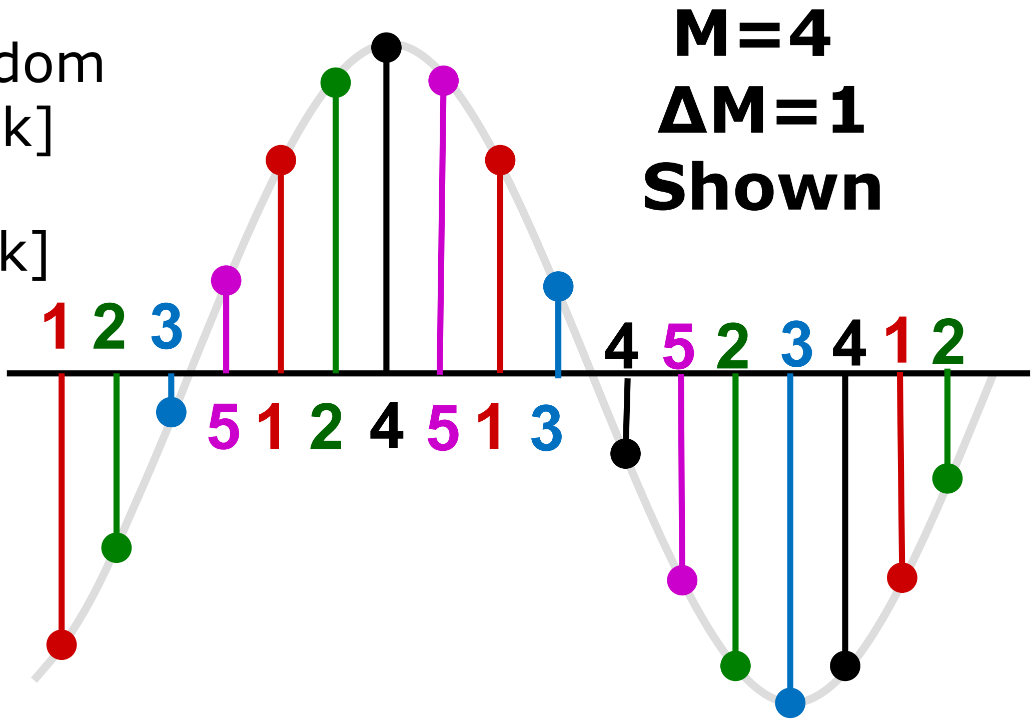


- □ Multiply V(kT) by pseudo-random sequence R=+/-1 to randomize sign
- □ Multiply ADC output by same sequence R to restore sign ($R^2=1$)
  - Randomizes sign of offset O_n in D[k] => No longer periodic => No spurs
  - Gain/Timing mismatch unaffected

Random Slice-Order Shuffling [6]

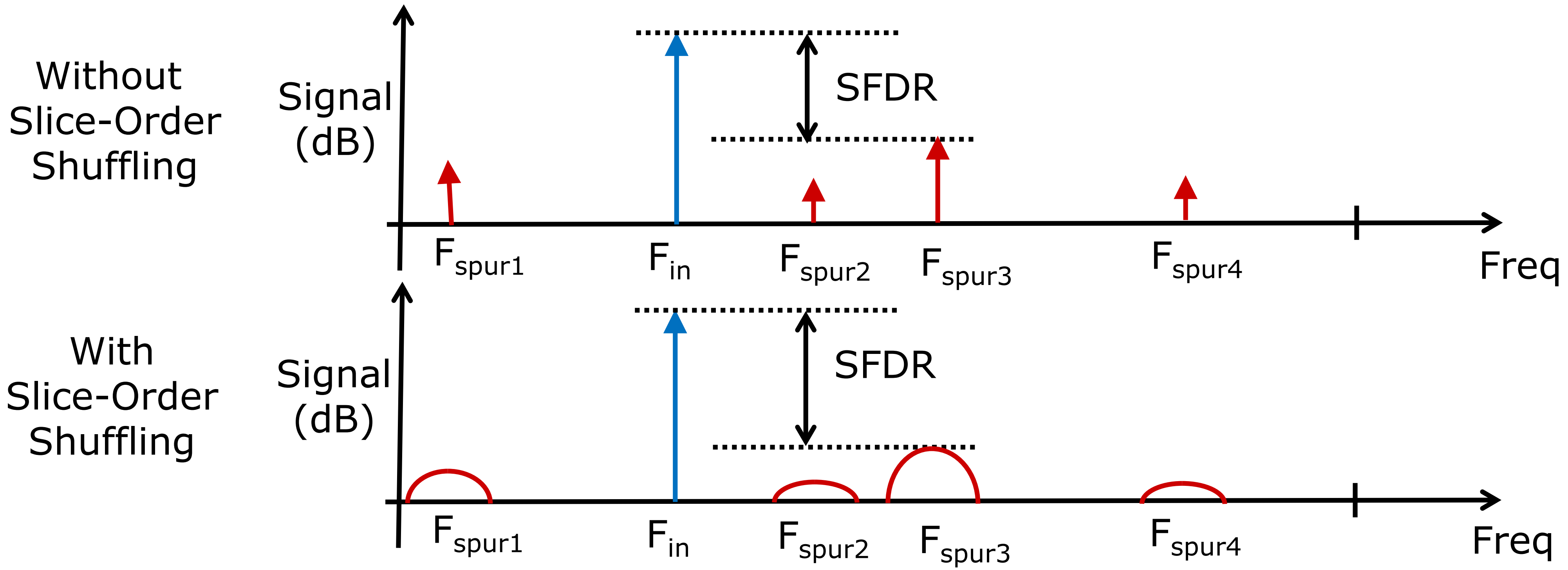


Random Interleave



- □ Interleave M+ΔM ADC slices, each with MT conversion time
  - 1+ ΔM ADC slices available at some sample times => Extra slices needed
  - Randomize which slice is used based on pseudo-random sequence R[k]
  - Interleave data using same sequence R[k] to get correct sample order D[k]

Random Slice-Order Shuffling



- Randomizing slice order spreads out spurs in Frequency
  - Improves SFDR for offset, gain and timing mismatch spurs
  - Larger ΔM => More ADC choices => More spur reduction
- Adds a lot of complexity to clock path implementation

Outline

- □ Background and Motivation
- □ Error Sources in Time-Interleaved ADCs
- □ Frequency Spectrum of Interleave Errors
- □ Interim Q+A
- □ Correction of Interleave Errors
- □ Interleaved ADC Architectures.
- □ Summary and Conclusions

Summary

- Time-Interleaved ADCs are attractive for high sample rate applications
- Mismatch in ADC slices limits SNDR performance
  - Concentration of errors as spurs also limits SFDR
- Correction techniques exist to address these errors
- Front-rank T/H improve bandwidth and reduces timing mismatch errors
- Randomization techniques can spread mismatch errors in frequency to improve SFDR

ISSCC 2020 – Papers of Interest

Suggested papers based on Advance Program:

- Session 16
  - Paper 16.1 is an 18GS/s 12-bit ADC
    - 8-way time-interleaved pipelined ADCs
    - 2-way time-interleaved front-rank T/H with optional randomization
  - Paper 16.2 is a 10GS/s 8-bit ADC
    - 4-way time-interleaved time-domain ADCs
- Session 6
  - Paper 6.1 is a 112Gb/s PAM-4 Wireline Transceiver
    - Uses a 36-Way time-interleaved 56GS/s SAR ADC

Further Reading

- □ Other tutorials and overviews:
  a) B. Murmann, "A/D converter circuit and architecture design for high-speed data communication," Proceedings of the IEEE 2013 Custom Integrated Circuits Conference, San Jose, CA, 2013, pp. 1-78
  b) A. Buchwald, "High-speed time interleaved ADCs," in IEEE Communications Magazine, vol. 54, no. 4, pp. 71-77, April 2016
  c) K. C. Dyer et al., "Calibration and Dynamic Matching in Data Converters: Part 2: Time-Interleaved Analog-to-Digital Converters and Background-Calibration Challenges," in IEEE Solid-State Circuits Magazine, vol. 10, no. 3, pp. 61-70, Summer 2018
- □ These tutorials provide good overviews of this topic
  - ■ References b) and c) also provide a comprehensive list of primary references

Key References

1. B. Murmann, "ADC Performance Survey 1997-2019," [Online]. Available: http://web.stanford.edu/~murmann/adcsurvey.html
2. W.C. Black and D.A. Hodges,"Time-interleaved converter arrays," IEEE J. Solid-State Circuits, vol. 15, no. 12, pp. 1022–1029, Dec. 1980
3. K. Poulton et al., "A 1-GHz 6-bit ADC System," IEEE J. Solid-State Circuits, vol. SC-22, no. 6, pp. 962– 970, Dec. 1987
4. C. Conroy, "A High-Speed Parallel Pipeline A/D Converter Technique in CMOS", Ph.D. Dissertation, University of California, Berkeley, 1994
5. S.M. Jamal et al., "A 10-b 120-Msample/s time-interleaved analog-to-digital converter with digital background calibration," IEEE J. Solid-State Circuits, vol. 37, no. 12, pp. 1618–1627, Dec. 2002
6. M. Tamba et al., "A method to improve SFDR with random interleaved sampling method," in Proc. Int. Test Conf., 2001, pp. 512–520