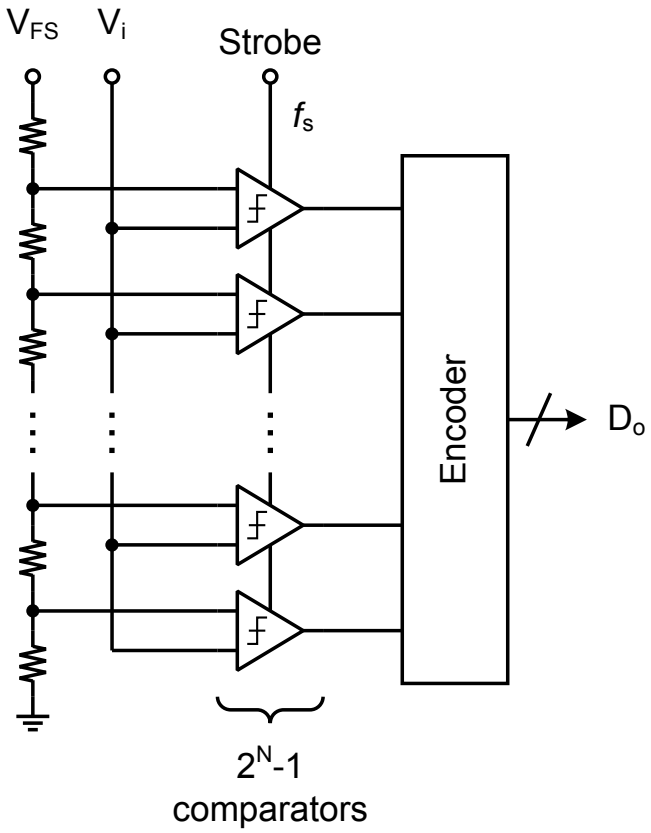
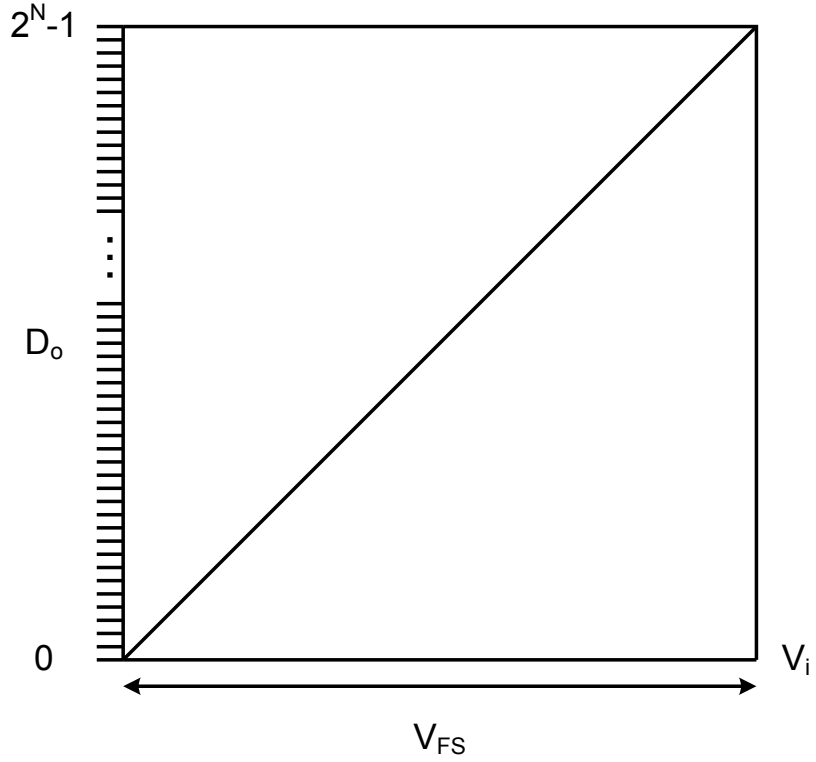
Subranging, Folding and Interpolation ADCs

Vishal Saxena, Boise State University
(vishalsaxena@boisestate.edu)

Subranging ADCs

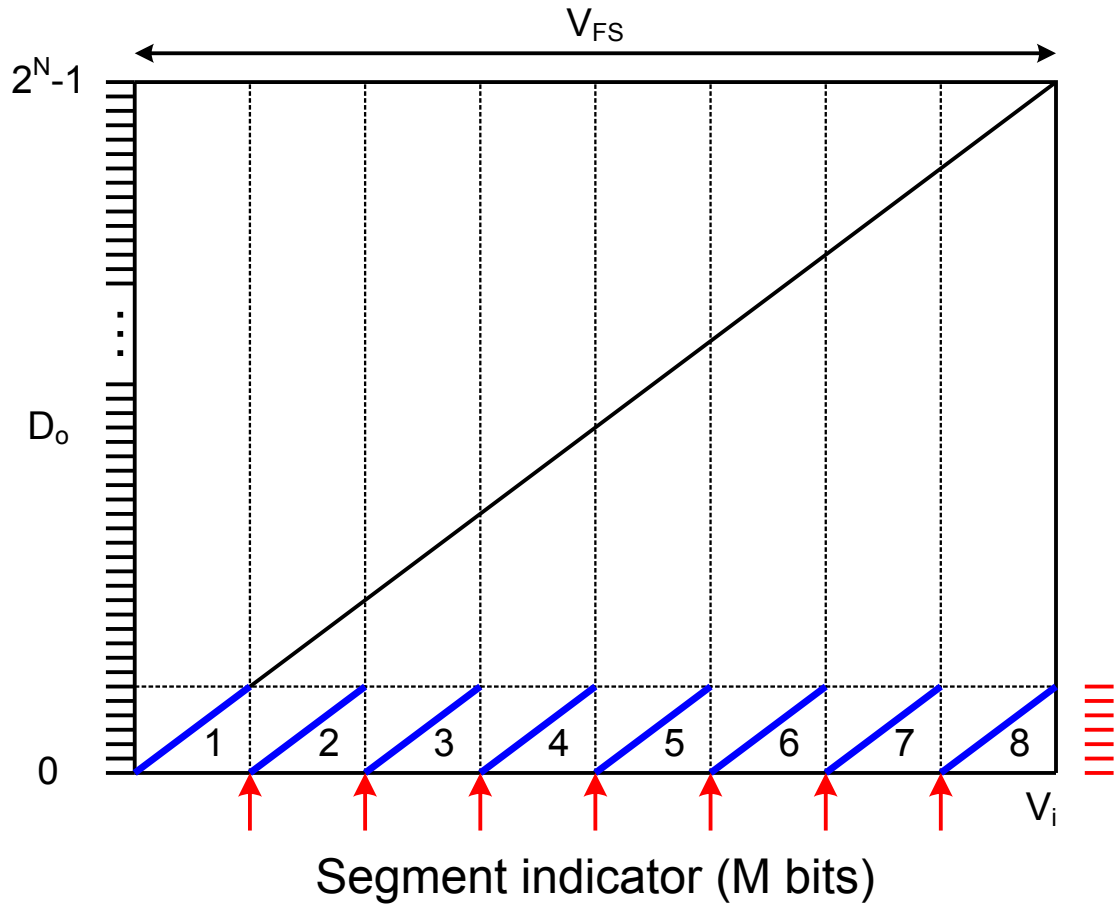
Inefficiency of Flash ADC





Only comparators in the vicinity of V_{in} are active at a time → low efficiency.

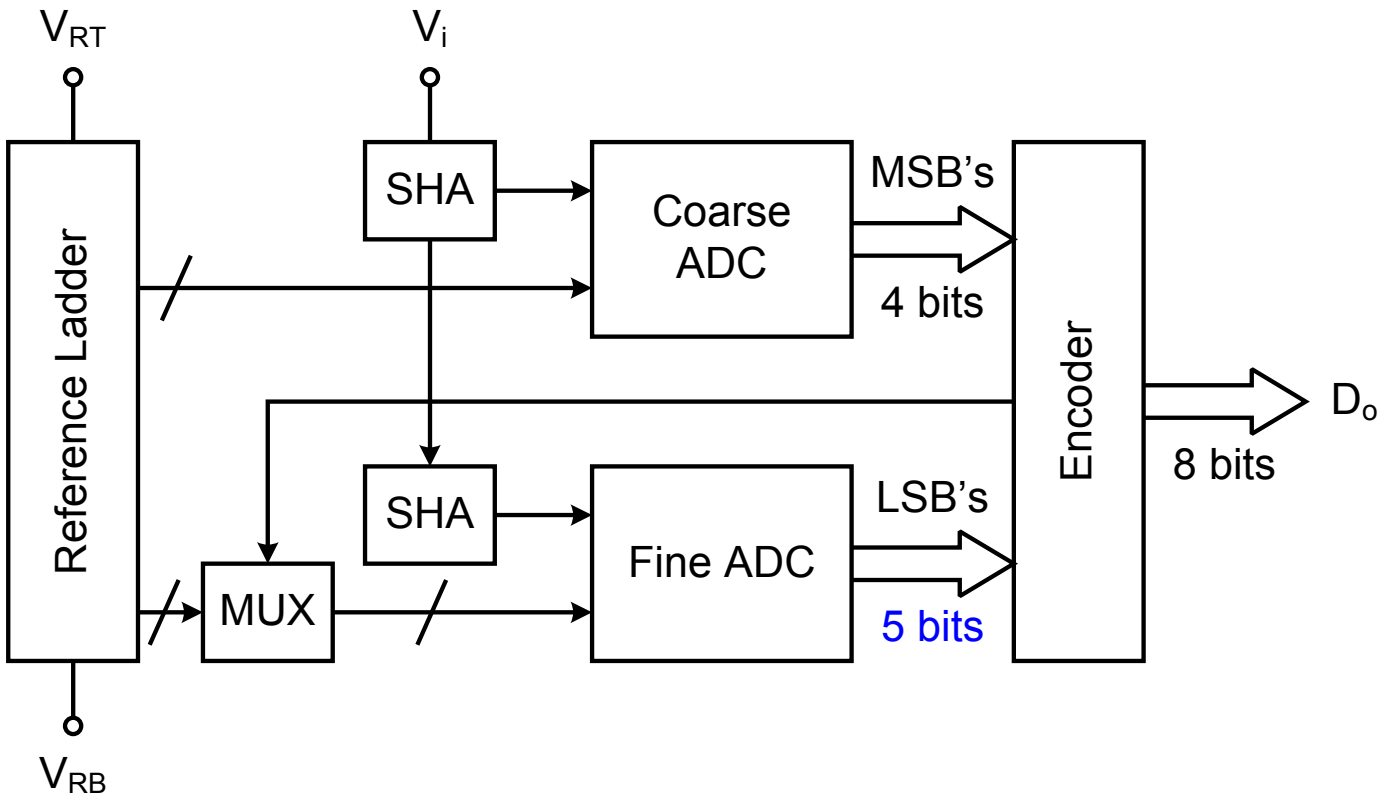
Segmented Quantization



Analog pre-processing divides V_{in} into 2^M uniformly-spaced segments.

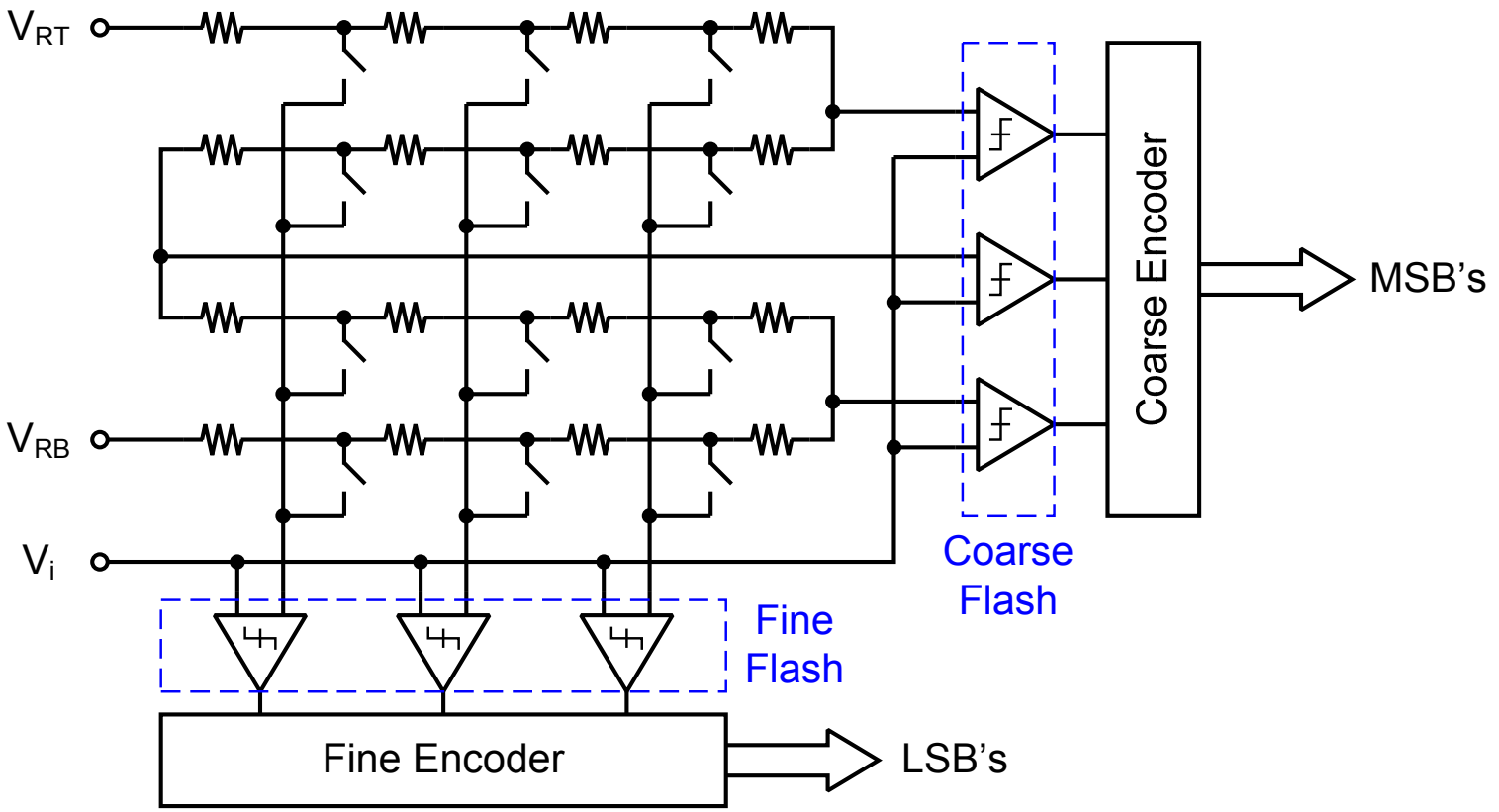
⇐ Fine quantization (N-M) bits

Typical Subranging Block Diagram



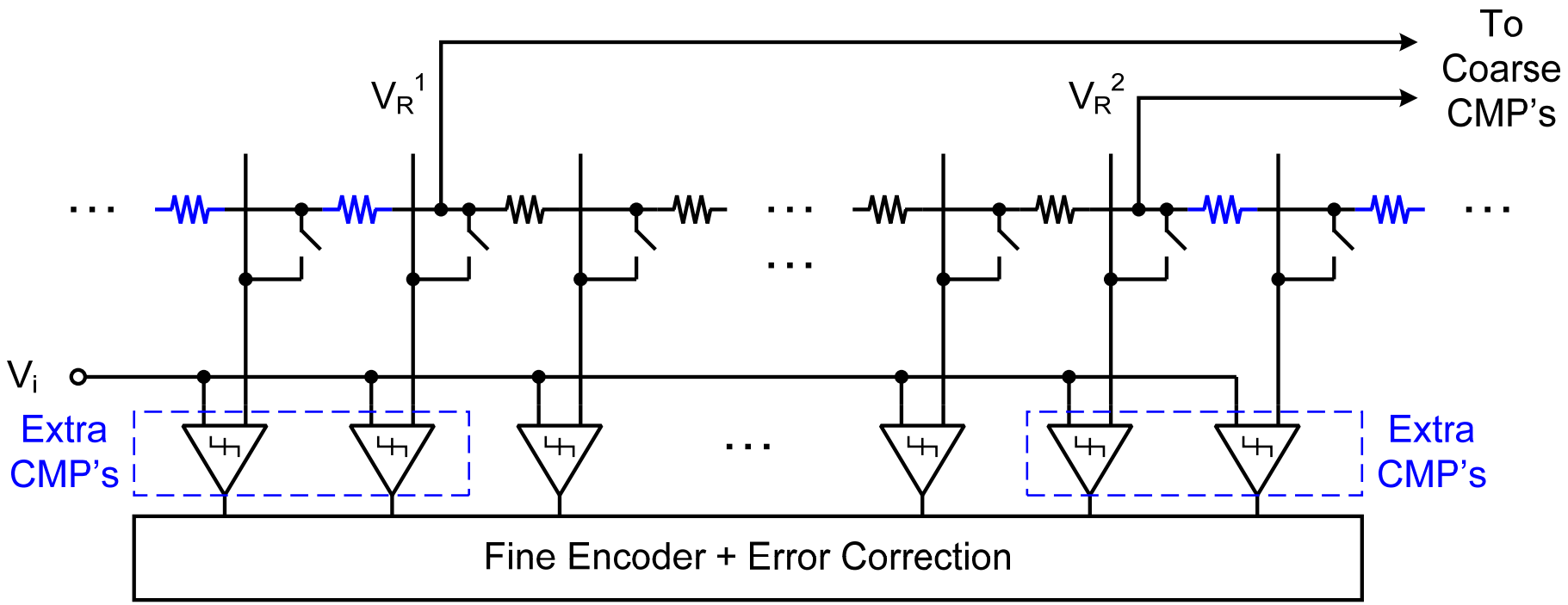
Redundancy in fine ADC provided by over- and under-range comparators

Subranging ADC Architecture



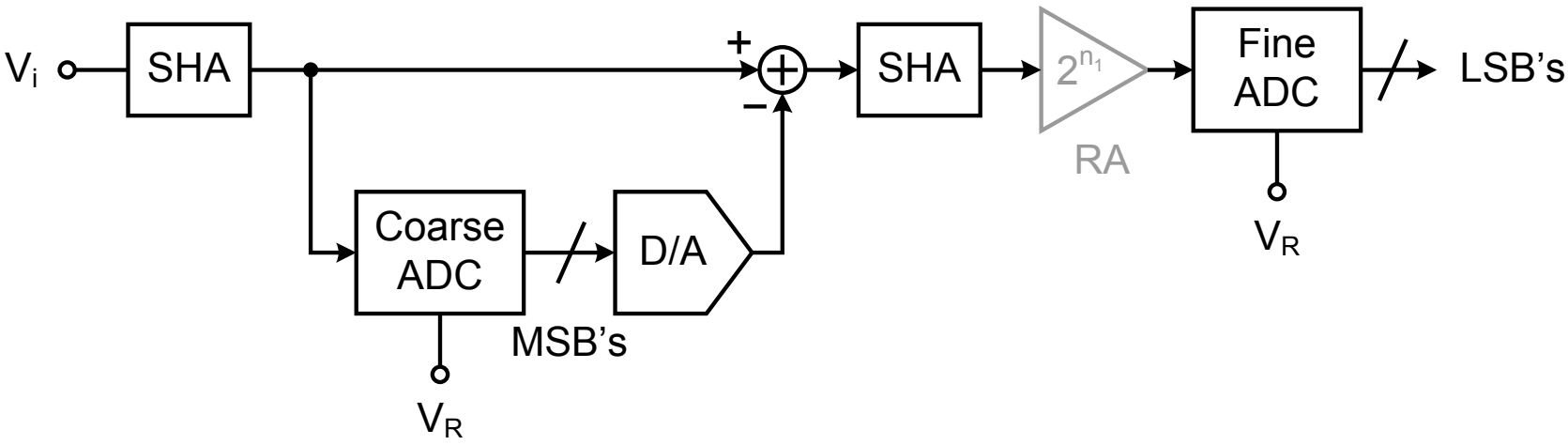
- Coarse comparators are connected to the coarse reference ladder taps
- When the MSBs are obtained, the fine ladder taps are addresses and then fine conversion takes place

Digital Redundancy in Fine ADC



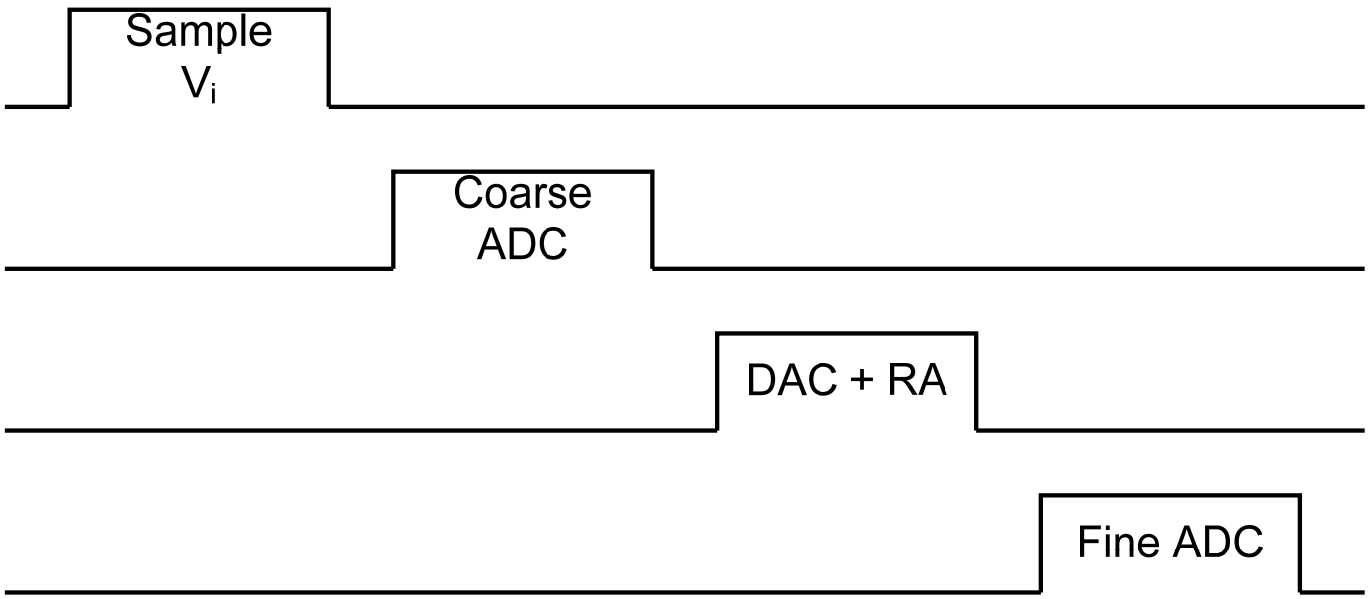
- The range of fine search extended on both sides (over and under ranging)
  - Corrects for small errors encountered in coarse conversion step
- Fine conversion comparators are critical (PA+AZ)

Two-Step Subranging/Pipelined ADC



- Coarse-fine two-step subranging architecture
- Conversion residue produced instead of switching reference taps
- Residue gain can be provided to relax offset tolerance in fine ADC
- Very similar to the pipelined architecture

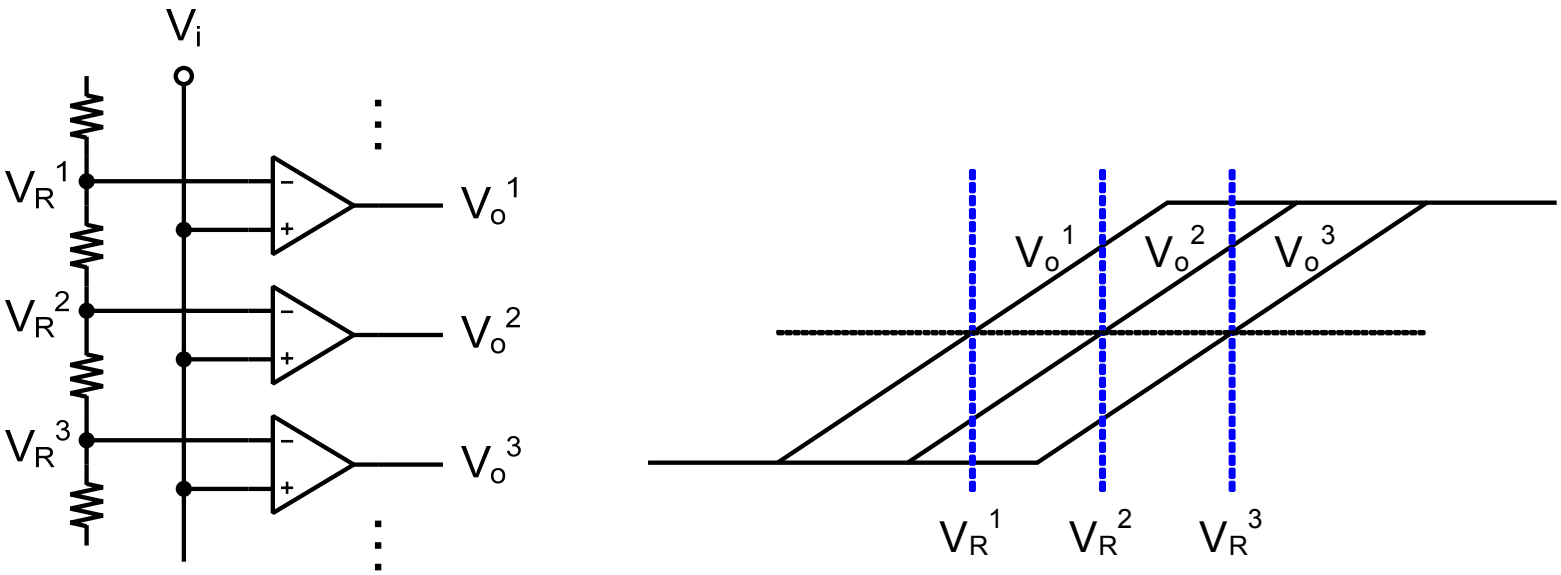
Timing Diagram



- Four conversion steps can be pipelined (needs op-amp)
- Usually DAC + RA settling consumes most of the conversion time
- Residue gain of unity is often used to speed up conversion

Interpolation

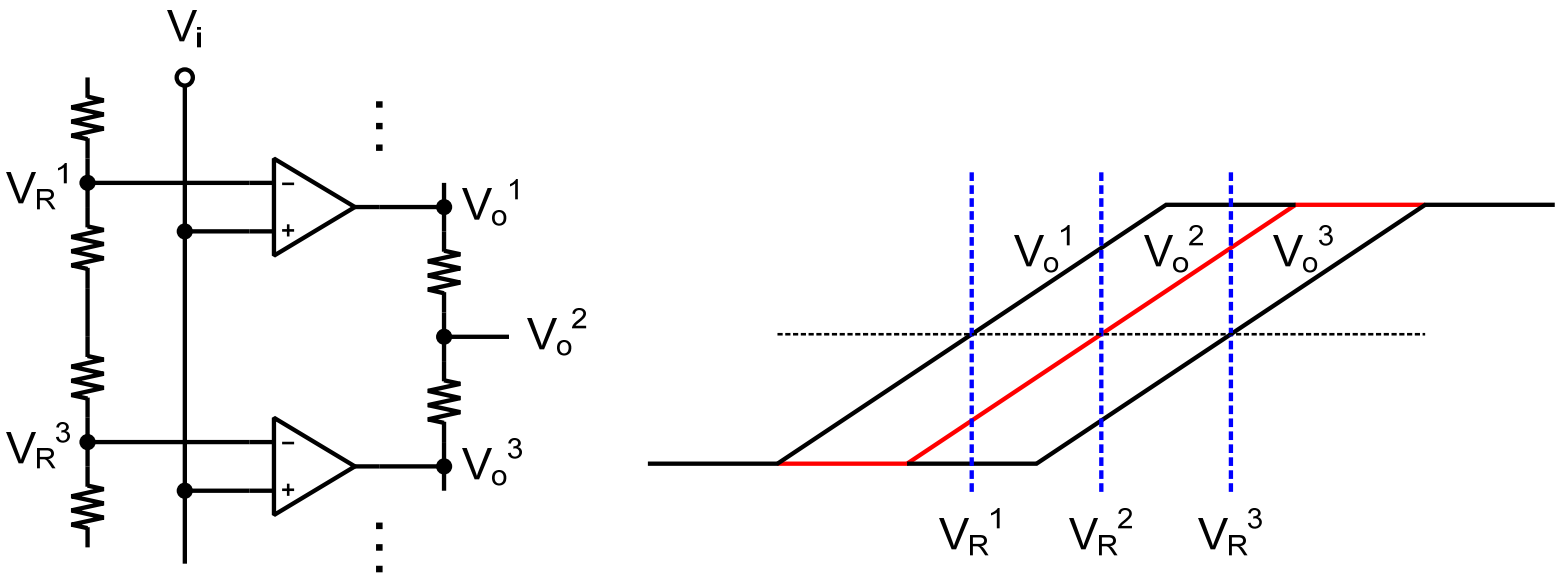
Preamp Interpolation



Uniformly spaced zero-crossings in flash ADCs (V_{out} vs V_{in})

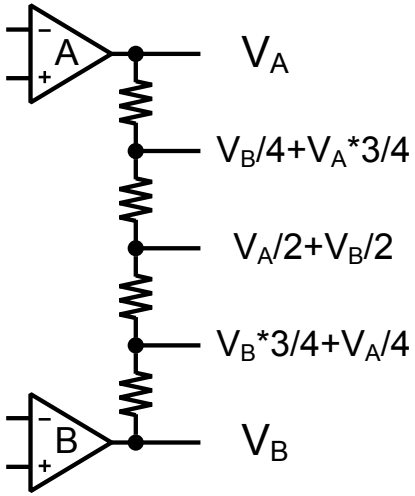
Ref: R. van de Grift, I. W. J. M. Rutten, and M. van der Veen, "An 8-bit video ADC incorporating folding and interpolation techniques," *IEEE Journal of Solid-State Circuits*, vol. 22, pp. 944-953, issue 6, 1987.

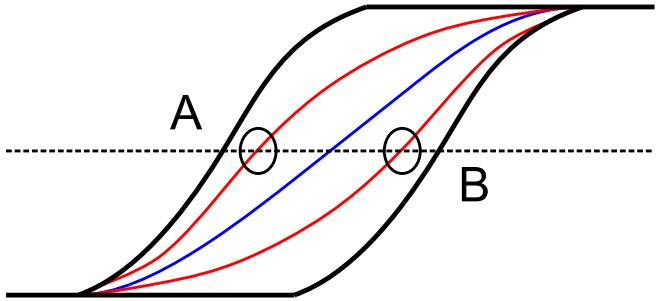
Resistive Interpolation



- Intermediate zero-crossings are recovered by interpolation.
- DNL is improved by voltage interpolation.
- Requires overlapped linear regions between adjacent preamps.

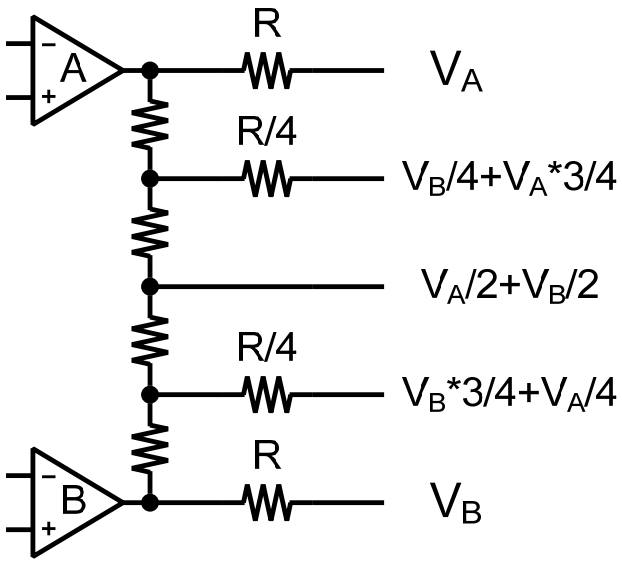
Interpolation Nonlinearity (I)





- Nonlinear TF of preamps cause errors in the interpolated zero-crossings.
- Interpolation nonlinearity directly translates into DNL and INL.
- Impedance mismatch due to interpolation also causes dynamic errors.

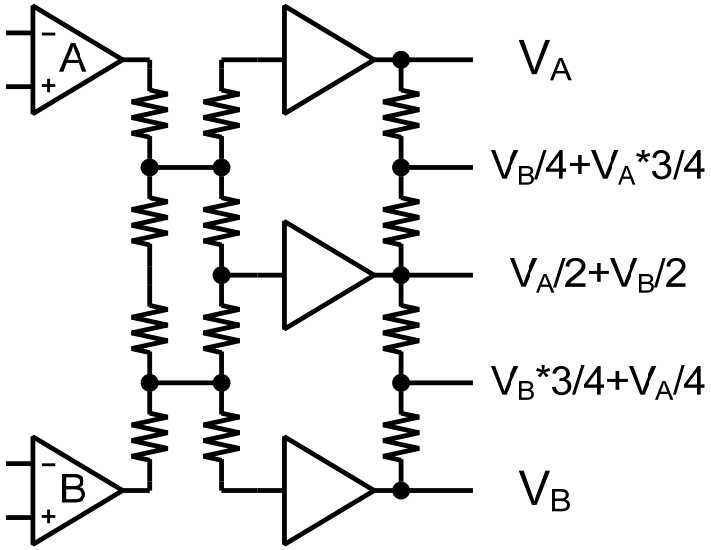
Interpolation Nonlinearity (II)



- Interpolation factor of 4
- Latch input bandwidth equalized
- Critical if input SHA is not used
- Nonlinear interpolation errors remain

Ref: R. J. van de Plassche and P. Baltus, "An 8-bit 100-MHz full-Nyquist analog-to-digital converter," *IEEE Journal of Solid-State Circuits*, vol. 23, pp. 1334-1344, issue 6, 1988.

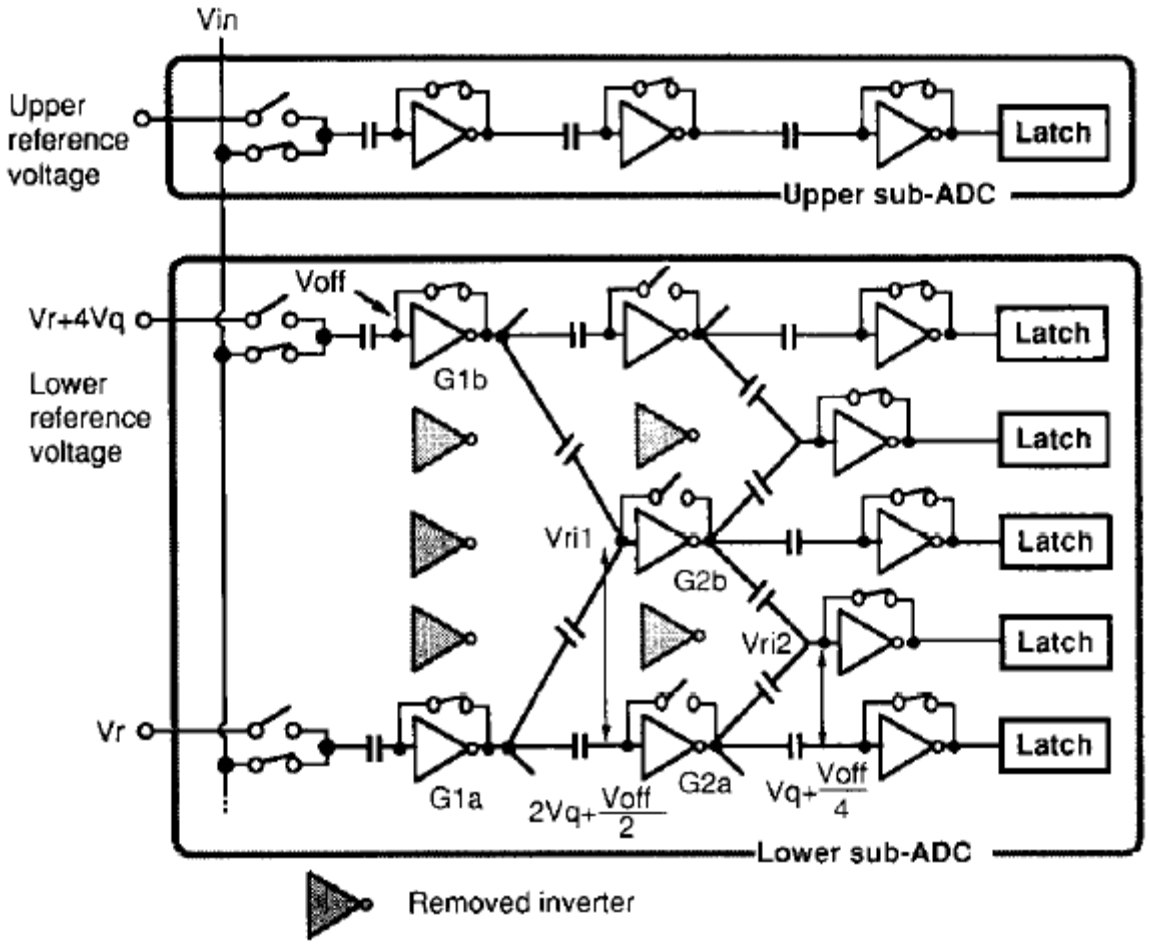
Interpolation Nonlinearity (III)



- Resistive mesh network improves impedance matching at latch input.
- 2-stage (2X) interpolation avoids the interpolation error.

Ref: P. Vorenkamp and R. Roovers, "A 12-b, 60-MSample/s cascaded folding and interpolating ADC," *IEEE Journal of Solid-State Circuits*, vol. 32, pp. 1876-1886, issue 12, 1997.

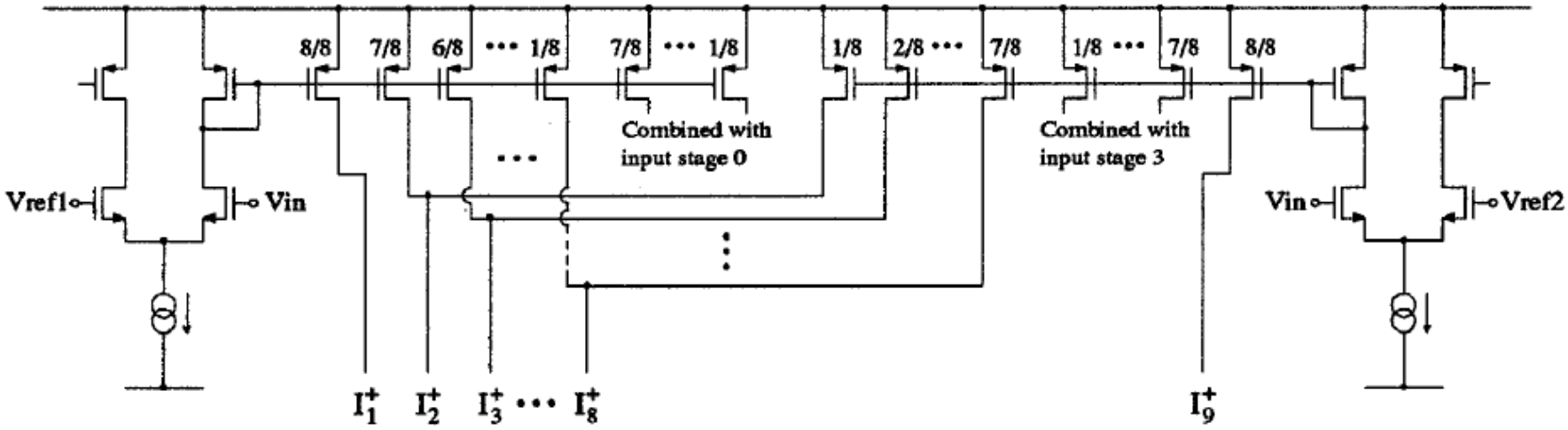
Capacitive Interpolation



K. Kusumoto,

JSSC, Dec. 1993

Current Interpolation



Less accurate then voltage interpolation due to mismatch of current mirrors

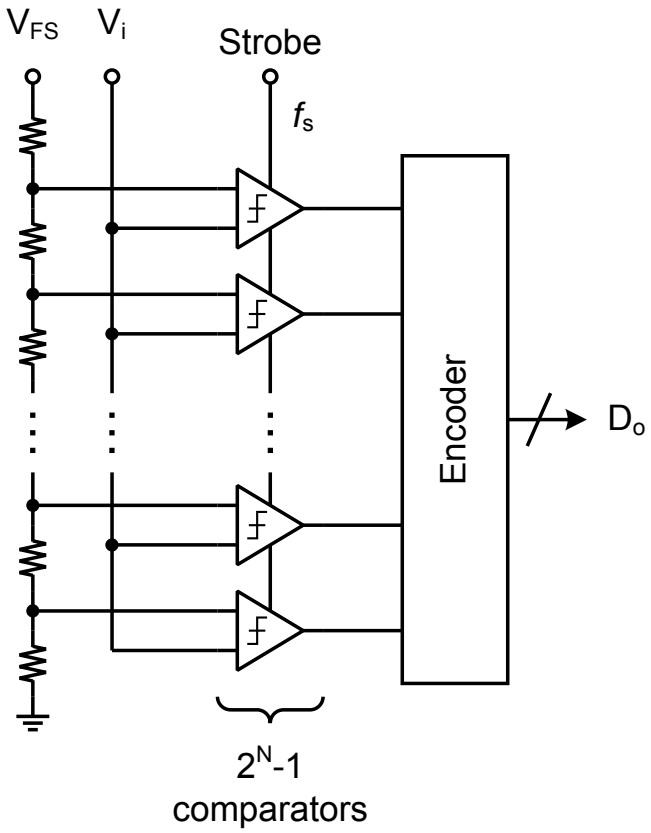
Ref: M. Steyaert, R. Roovers, and J. Craninckx, "100 MHz 8 bit CMOS interpolating A/D converter," in *Proceedings of IEEE Custom Integrated Circuits Conference*, 1993, pp. 28.1.1-28.1.4.

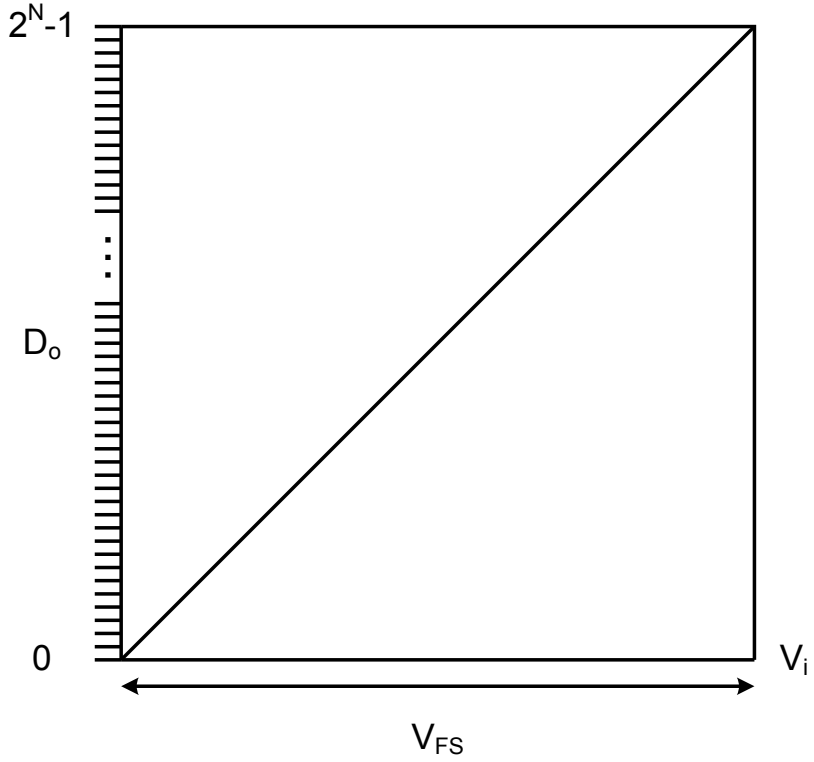
Features of Interpolation

- Reduces the total number of preamps by the interpolation factor
  - Reduces the total input capacitance (larger input BW)
- Total number of latches stay the same
- More area- and power-efficient than straight flash ADC
- Voltage interpolation improves DNL (error averaging)
- Loading of interpolation network decreases the preamp gain/BW
- Subject to preamp nonlinearities

Folding ADCs

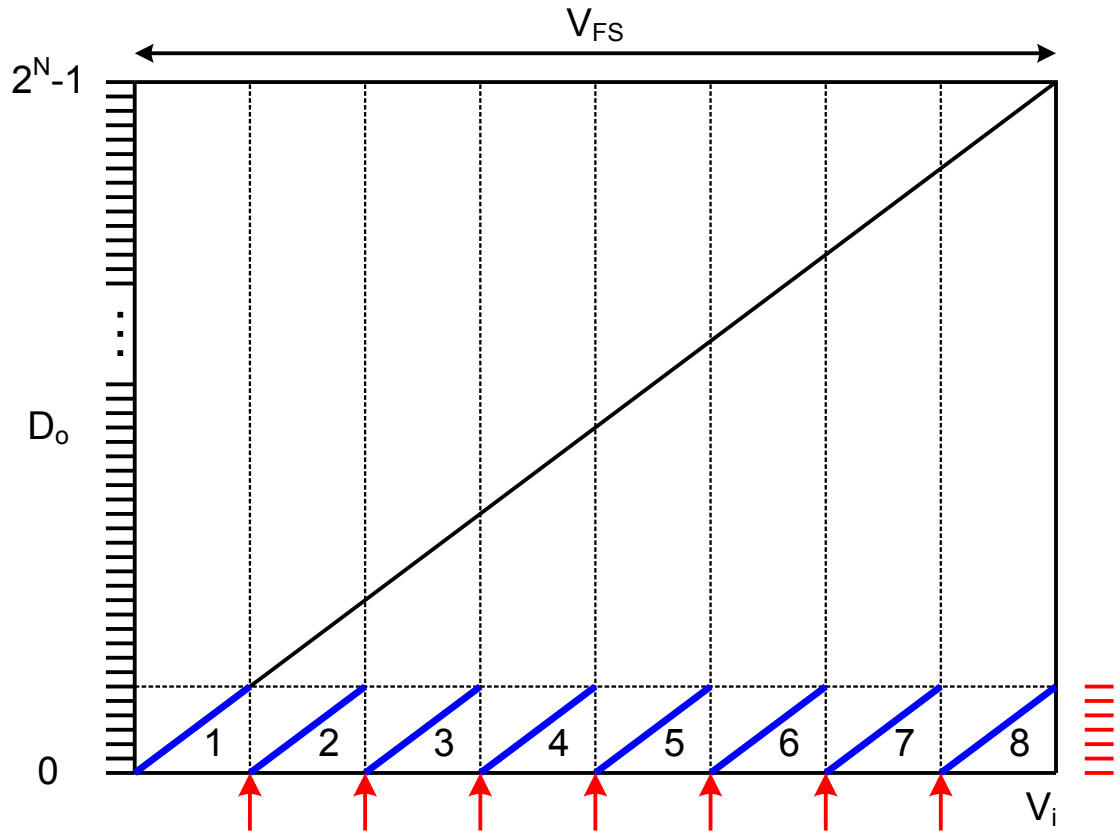
Inefficiency of Flash ADC





Only comparators in the vicinity of V_{in} are active at a time → low efficiency.

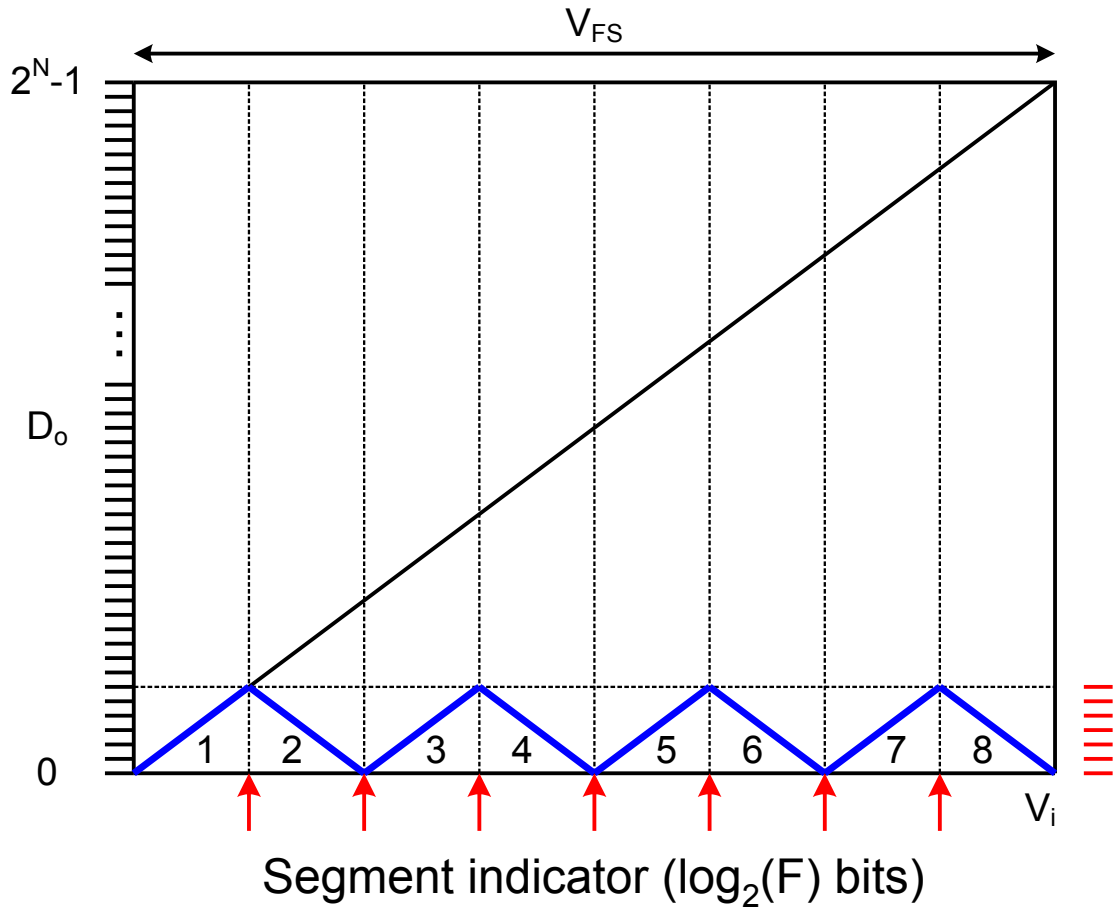
Segmented Quantization



Analog pre-processing divides V_{in} into 2^M uniformly-spaced segments.

⇐ Fine quantization (N-M) bits

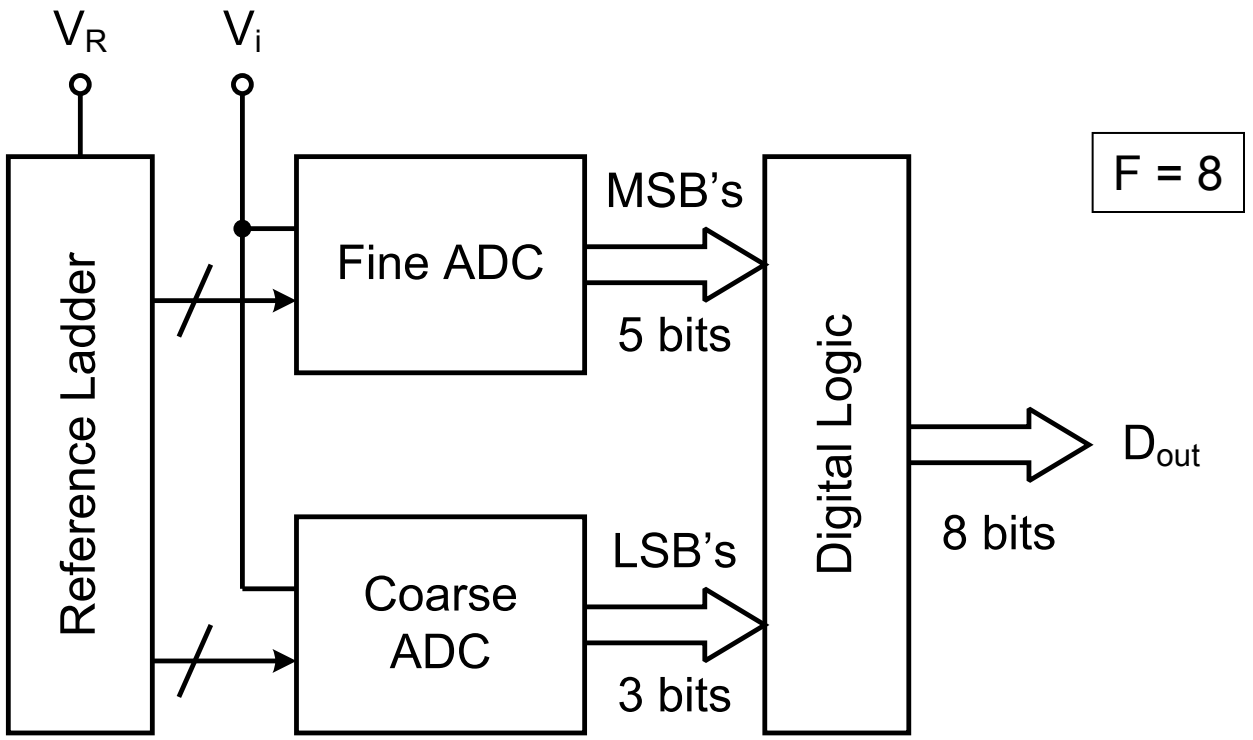
Signal Folding



- Analog pre-processing → folding amplifier
- Folding factor (F) is equal to the number of folded segments.

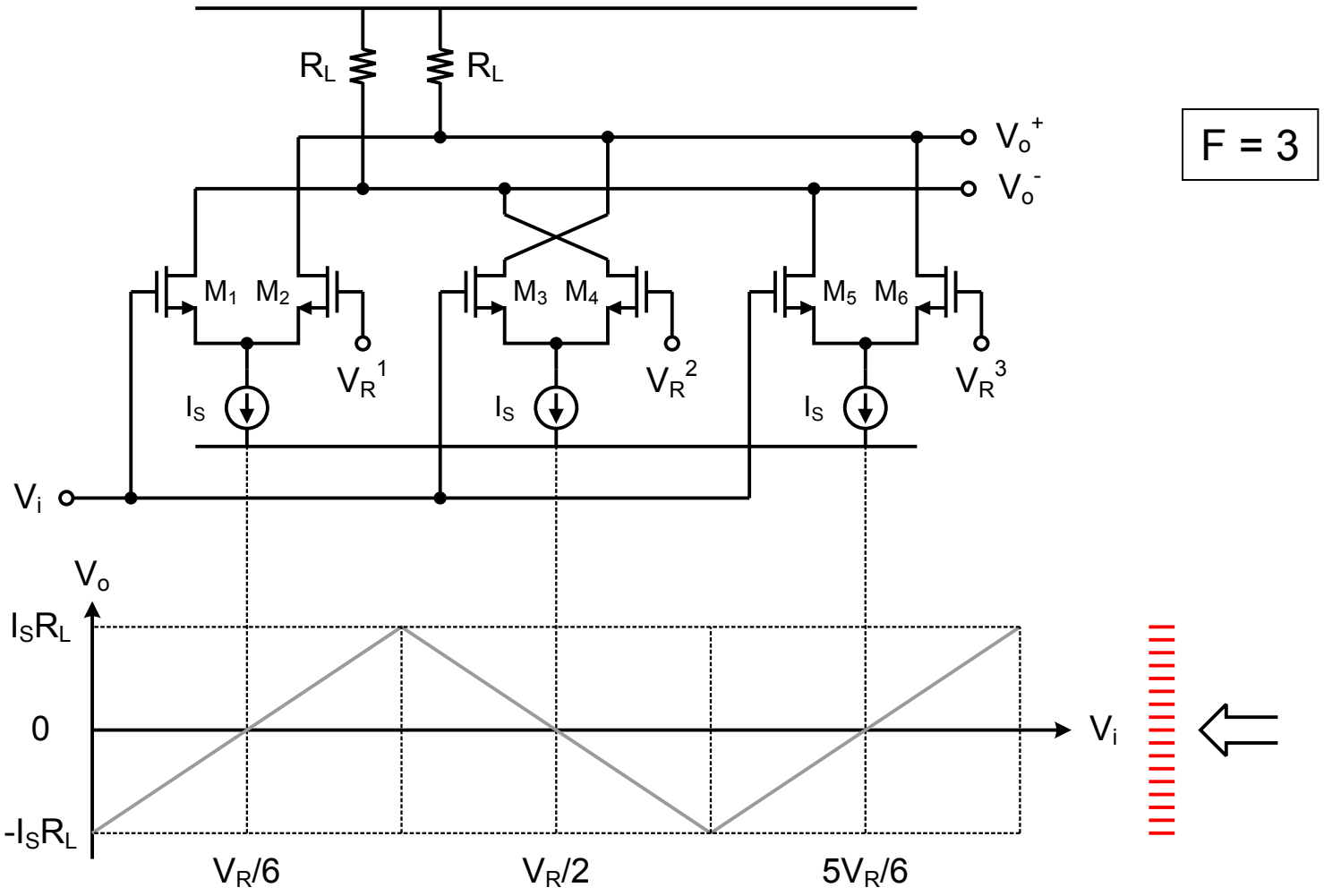
⇐ Fine quantization N-$\log_2$(F) bits

Folding ADC Architecture



- The fine ADC performs amplitude quantization on the folded signal.
- The coarse ADC differentiates which segment V_{in} resides in.

Folding Amplifier



Signal Folding

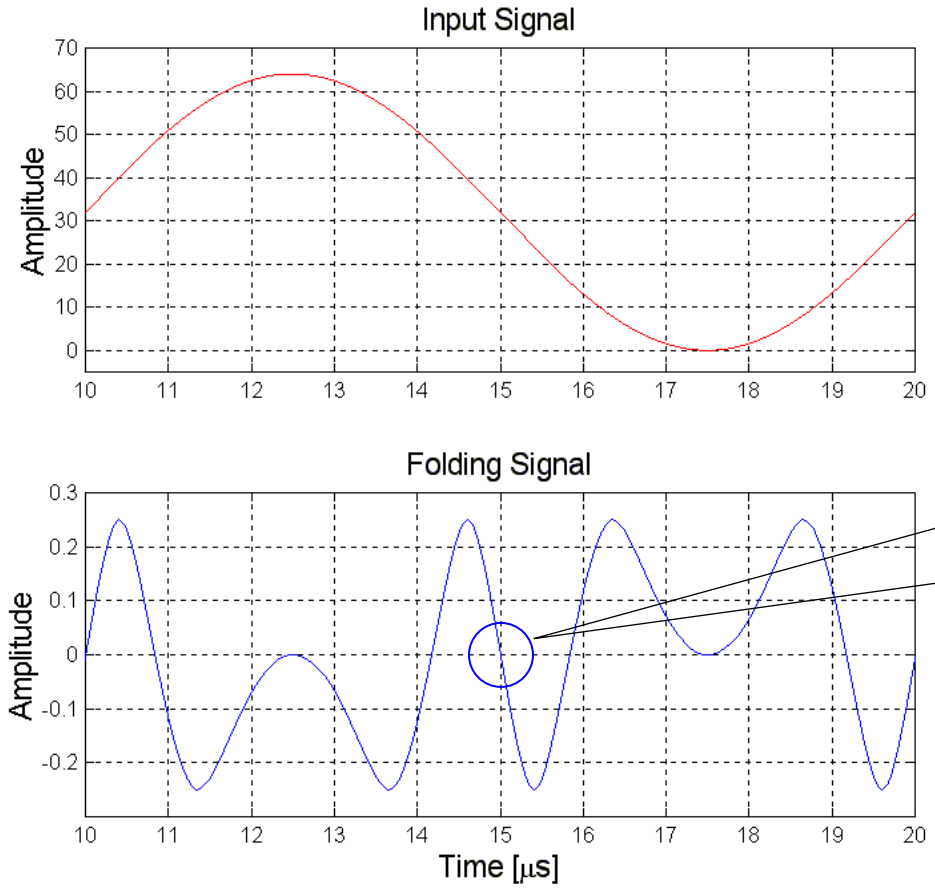
Pros

- Folding reduces the comparator number by the folding factor F, while the number of preamps remains the same.

Cons

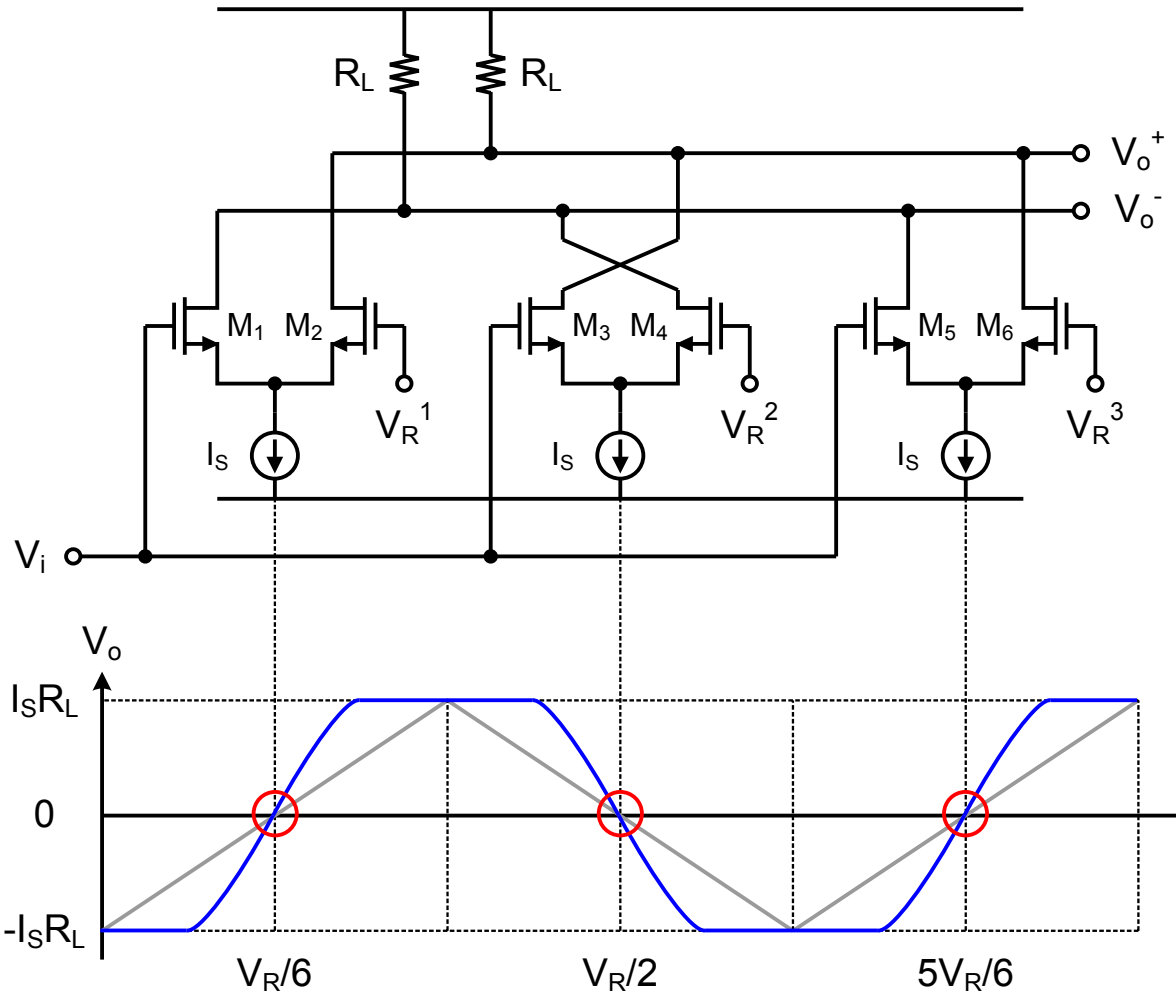
- Multiple differential pairs in the folder increases the output loading.
- “Frequency multiplication” at the folder output.

Frequency Multiplication



$$f_{max} = \frac{\pi \cdot f_{in}}{\sin^{-1}(\frac{2}{F})}$$

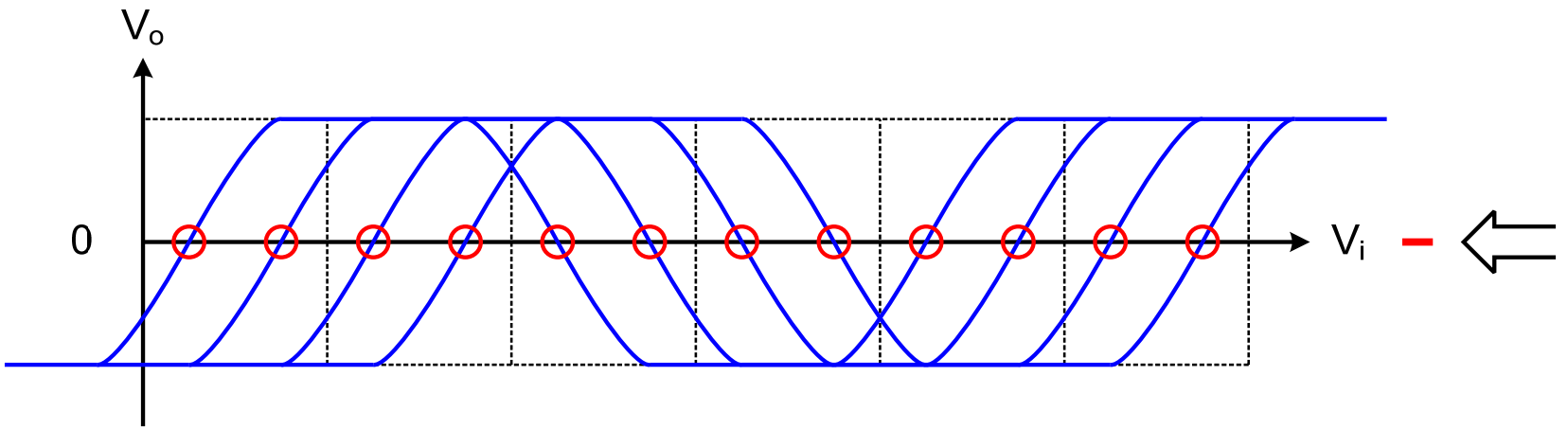
Folding Amplifier



F = 3

Zero-crossings are still precise!

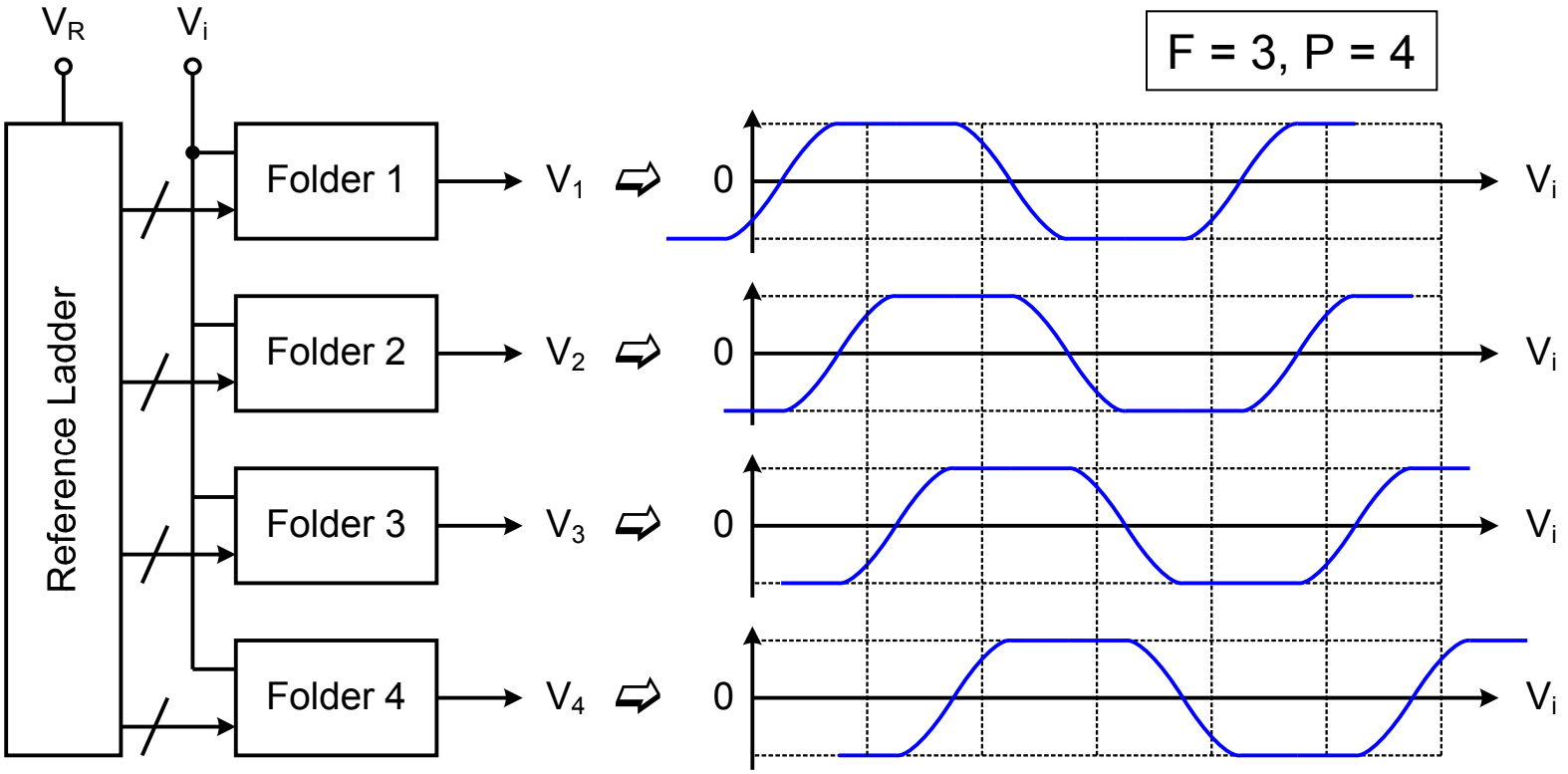
Zero-Crossing Detection



F = 3, P = 4

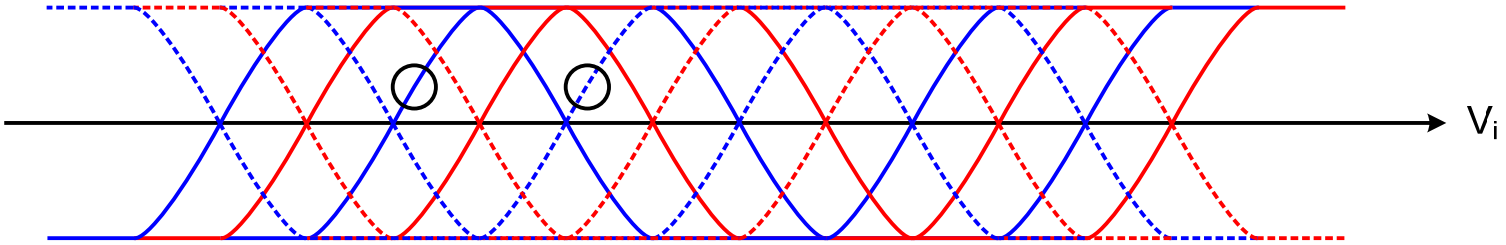
- Only detect zero-crossings instead of fine amplitude quantization → insensitive to folder nonlinearities.
- P parallel folding amplifiers are required.

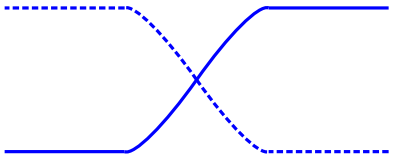
Offset Parallel Folding



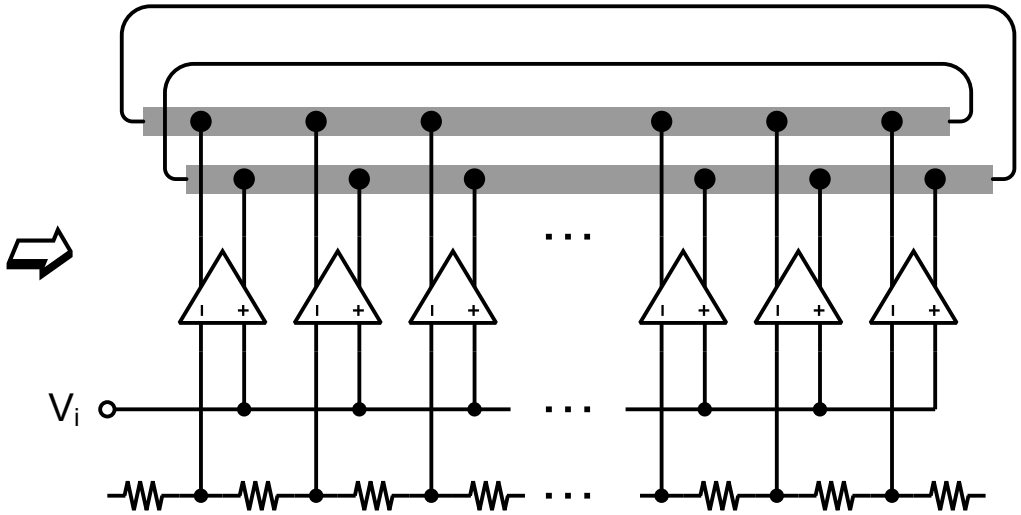
- Total # of zero-crossings = Total # of preamps = **P*F**
- Parallel folding saves the # of comparators, but not the # of preamps → still large C_{in}.

Folding + Interpolation

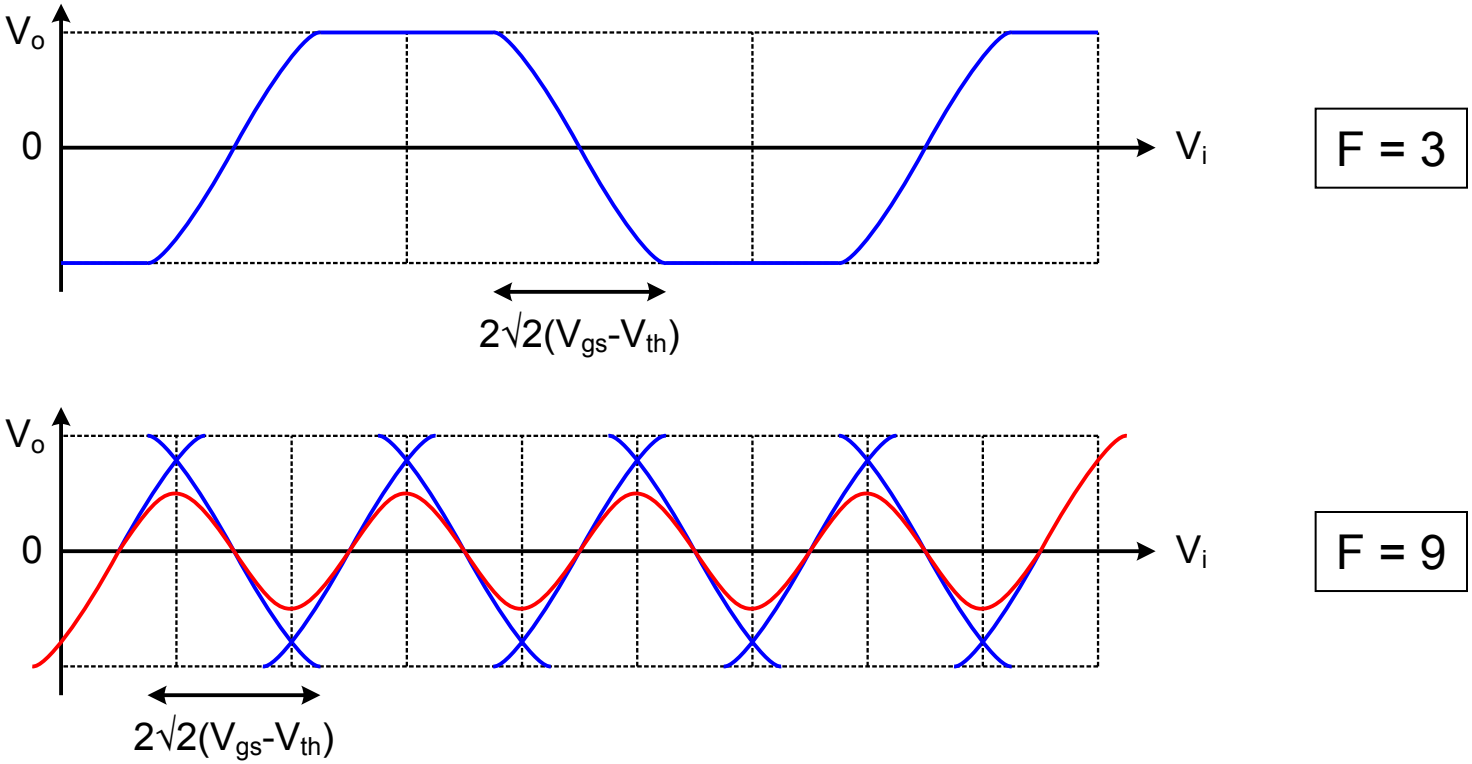




Cross-connect P & N sides at the endpoints



“Rounding” Problem



- Large F results in signal “rounding”, causing gain and swing loss.
- Max. folding factor is limited by V_{ov} of folder and supply voltage.

Folding vs. Interpolation

Folding

- Folding works better with **non-overlapped** active regions between adjacent folders.
- Large V_{ov} (for high speed) of folders and low supply voltage limit the max. achievable F.

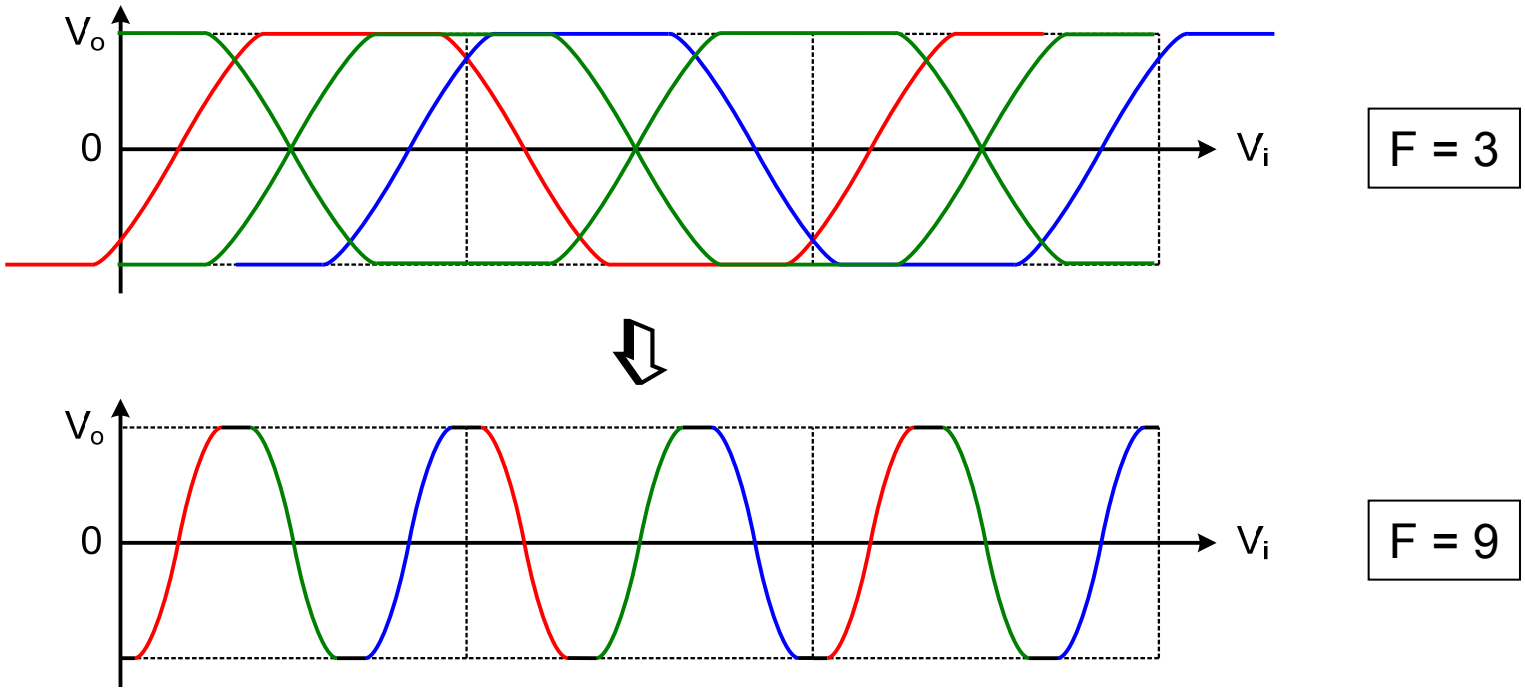
Interpolation

- Works better with closely spaced **overlapped** active region between adjacent folding signals.

Observation

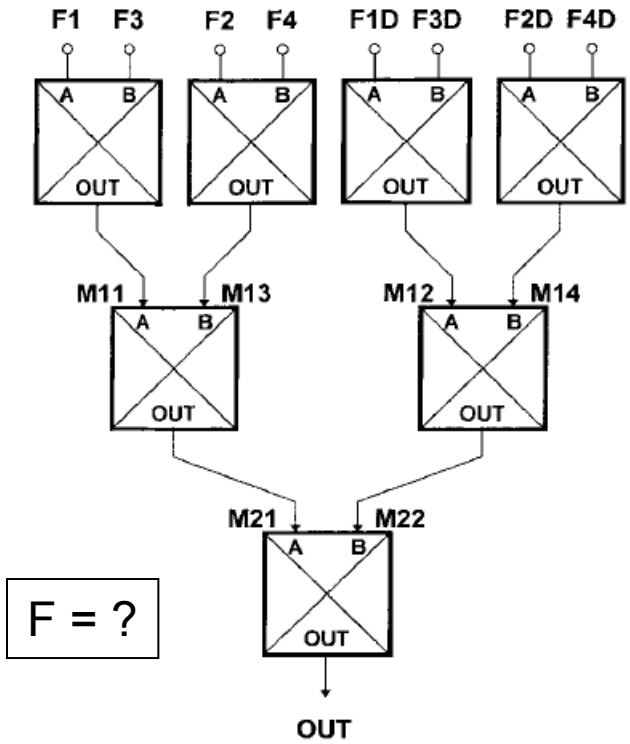
- Small F and large P (parallel folders) will help both folding and interpolation, but introduces large C_{in} – approaching flash…
- What else can we do?

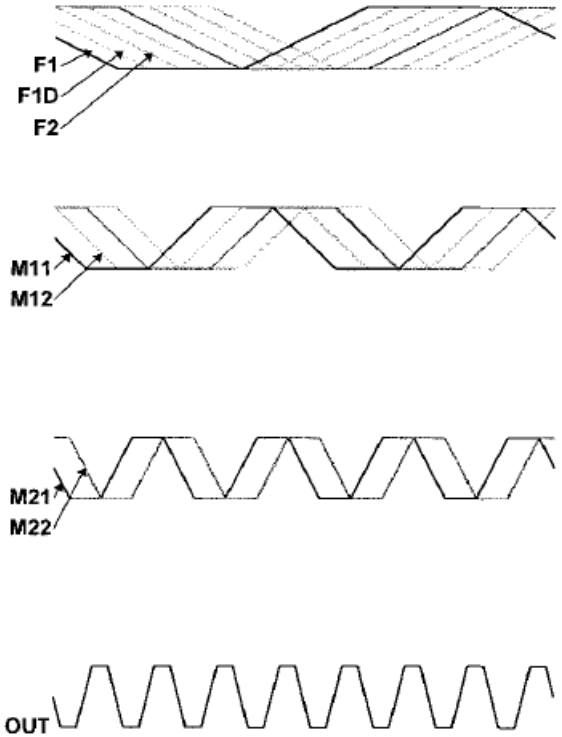
Cascaded Folding

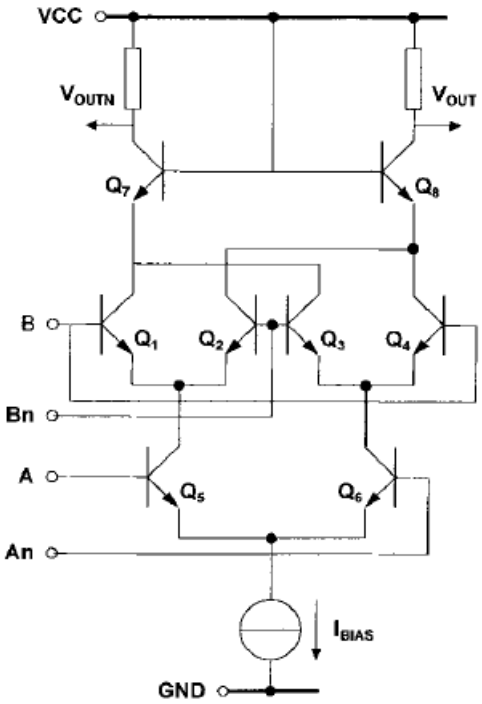


- Ideal: A large folding factor F can be developed successively.
- Small F in the 1st-stage folder → large V_{ov}, less capacitive loading, and less frequency multiplication effect.

Cascaded Folder Architecture (I)

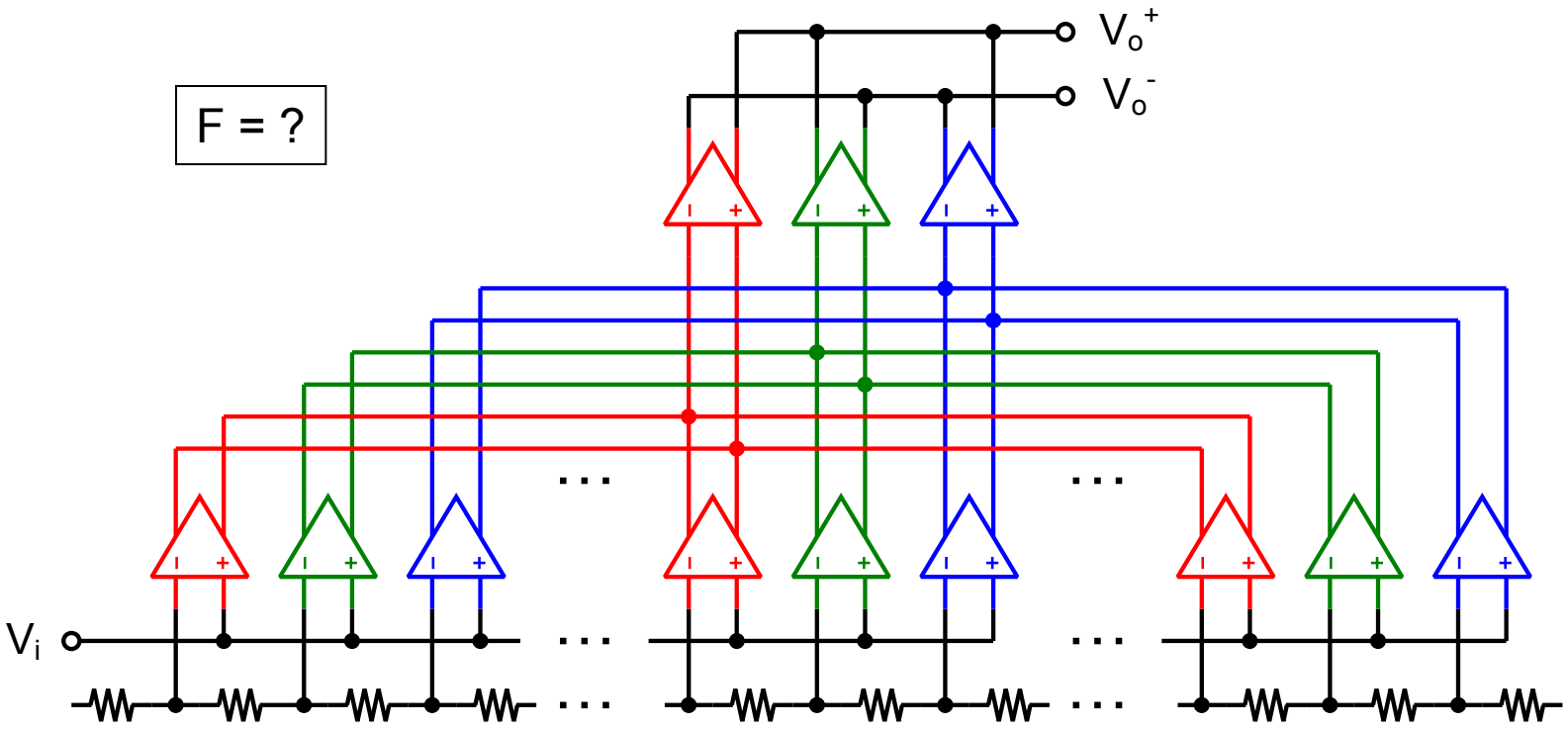






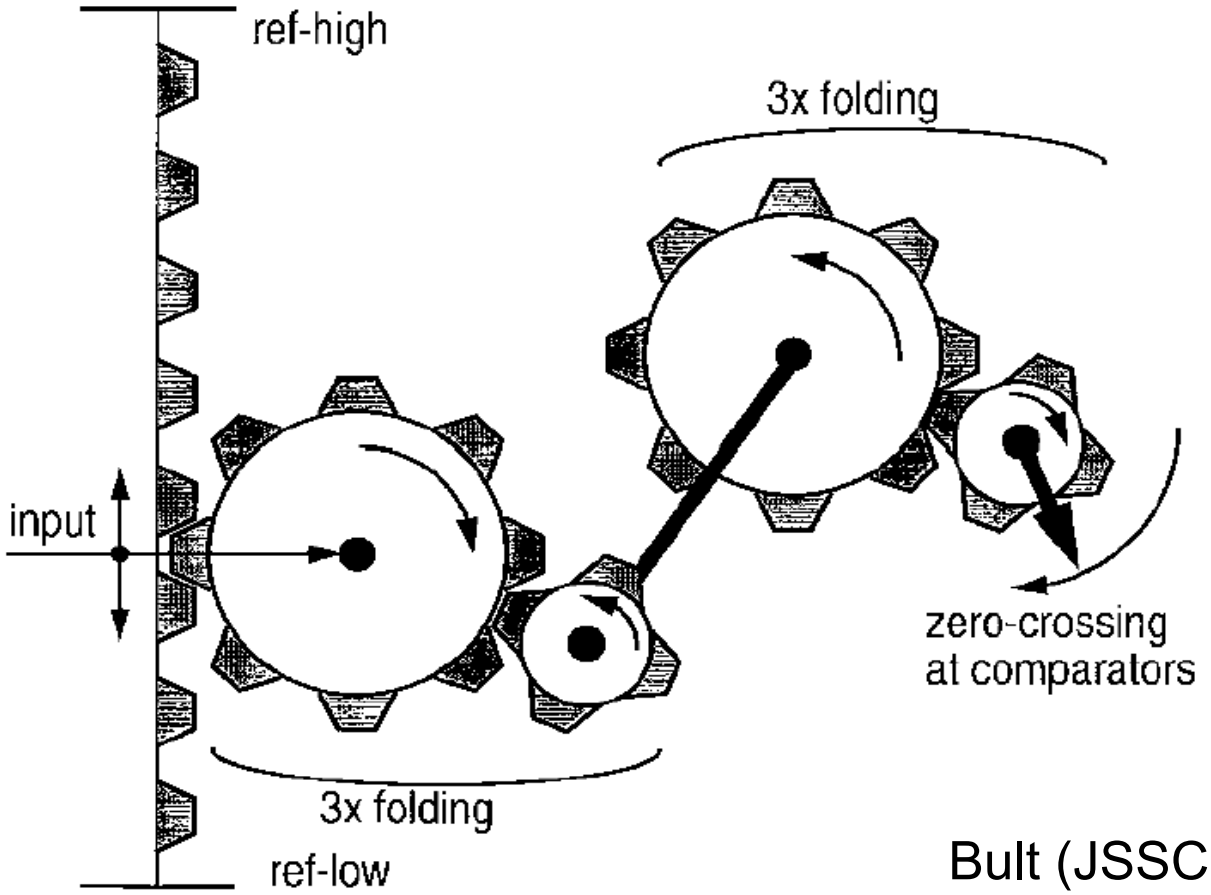
- Gilbert four-quadrant multiplier based folding amplifier
- Only works with even P, requires a lot of headroom

Cascaded Folder Architecture (II)



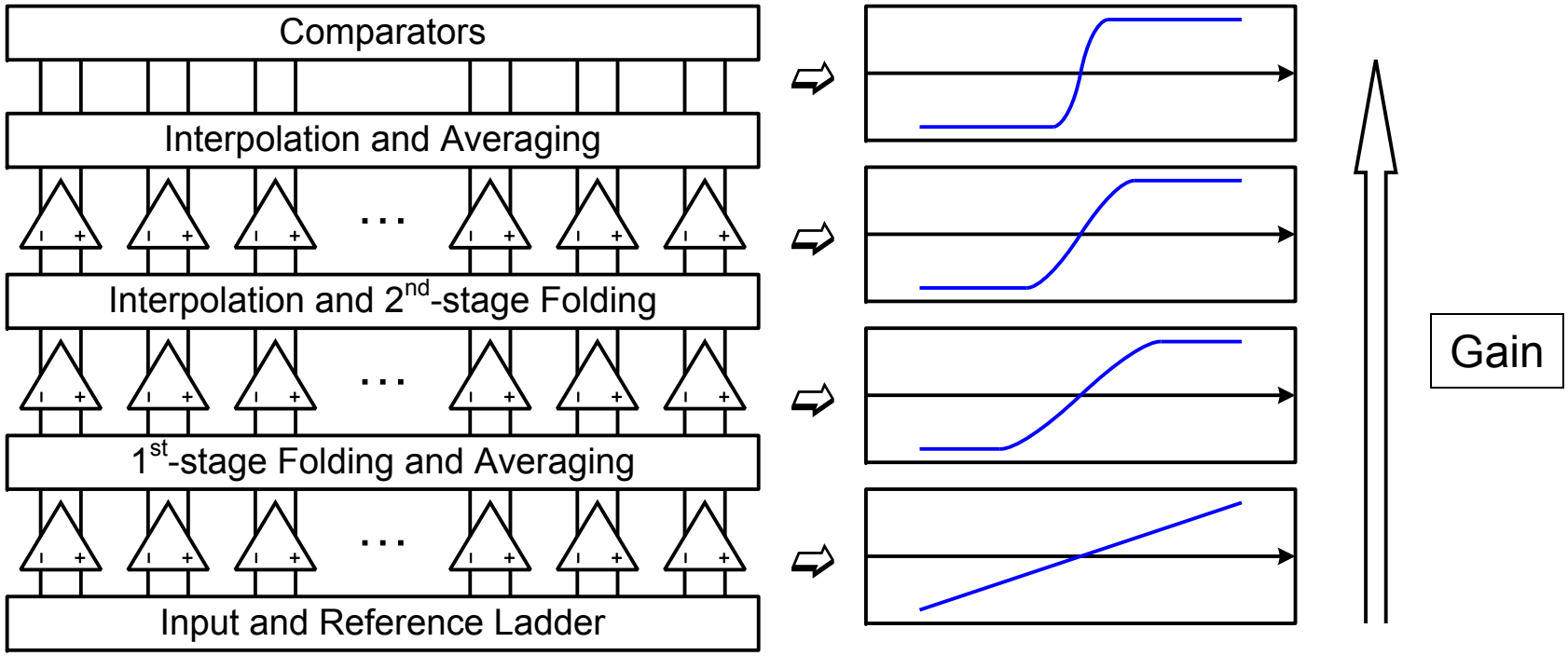
- Simple differential pair-based folding amplifiers
- Only works with odd P, compatible with low supply voltage.

Mechanical Model of Cascaded Folding



Bult (JSSC'97)

Distributed Preamplification



Large signal gain developed gradually along the signal path

→ from “soft” to “hard” decision

Useful Formulas

Assuming a two-stage cascaded folding & interpolating ADC,

F_1 = 1st-stage folding factor, F_2 = 2nd-stage folding factor,

P = # of offset parallel folders, I = total interpolation factor,

then,

Total # of decision level = $P*F_1*I$

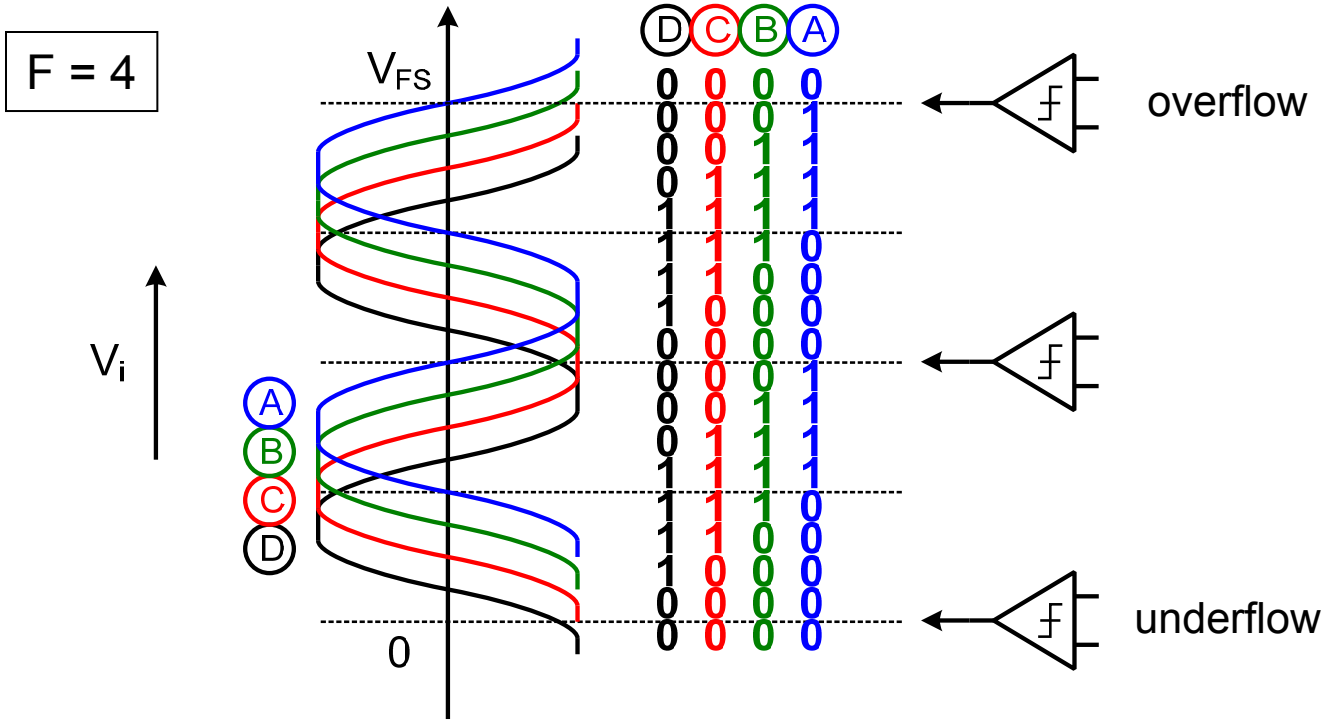
ADC Resolution = $\mathbf{Log_2(P*F_1*I)}$

Total # of fine comp = $P*I/F_2$

Total # of coarse comp = $P*F_1$ or $P+F_1$**?**

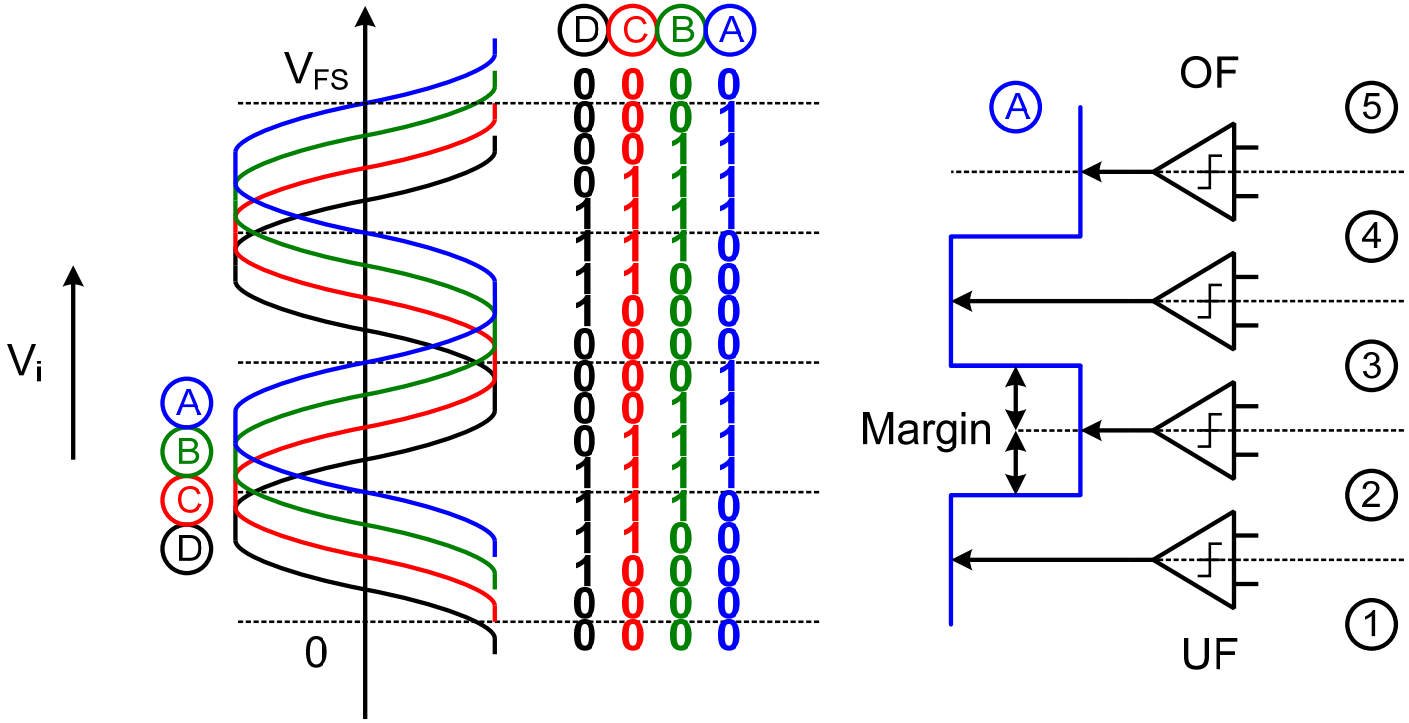
where, usually P = F_2 holds.

Circular Thermometer Code



- Very tight coarse comparator offset requirement (<1/2 LSB)
- Aperture delay b/t coarse and fine worsens the problem.

Coarse-Fine Sync (Bit Alignment)



- "Coarse" fold indicator with offset thresholds combined with the output of fine comparator A precisely determines which fold V_{in} resides in.
- Large tolerance on coarse comparator offset and aperture delay errors

Useful Formulas

Assuming a two-stage cascaded folding & interpolating ADC,

F_1 = 1st-stage folding factor, F_2 = 2nd-stage folding factor,

P = # of offset parallel folders ($P>F_2$), I = total interpolation factor, then

total # of decision level = $P*F_1*I$,

ADC Resolution = $Log_2(P*F_1*I)$,

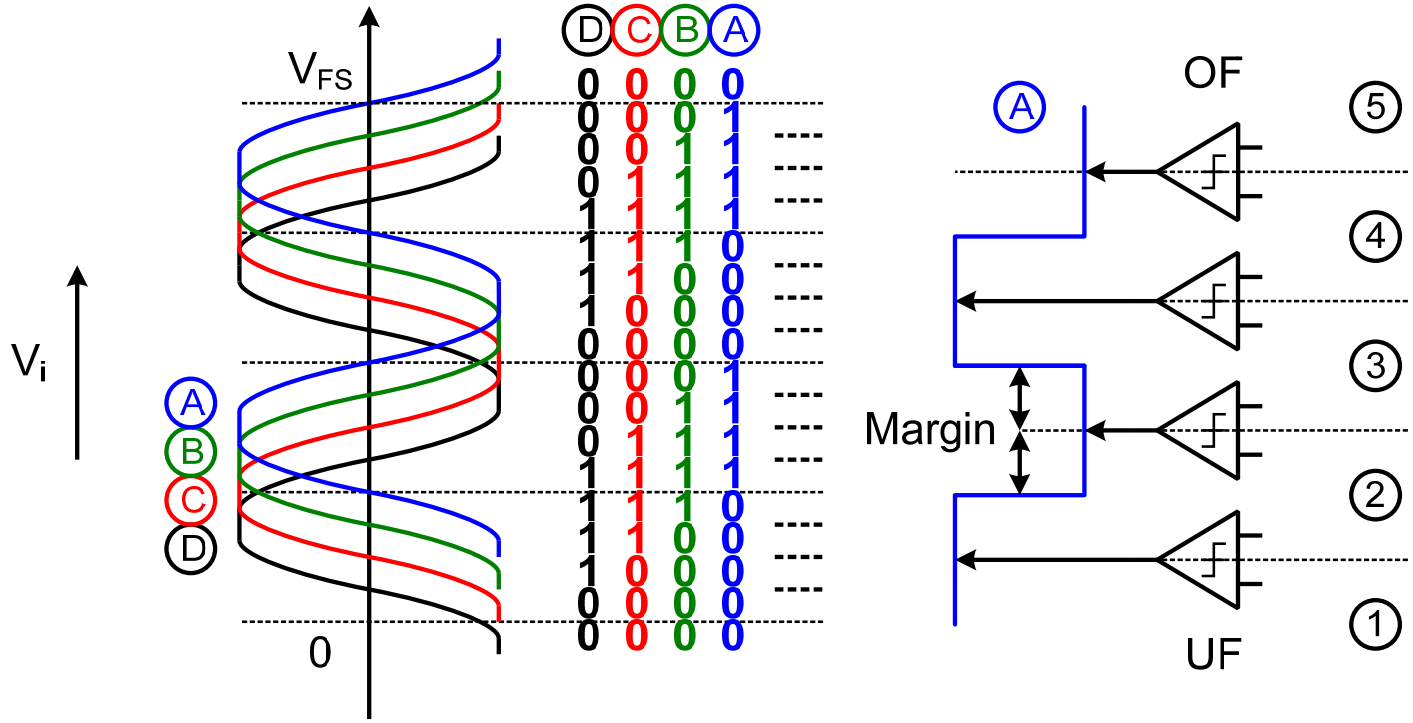
total # of preamps in 1st folder = $P*F_1$,

total # of preamps in 2nd folder = P,

total # of fine comparators = $P*I/F_2$,

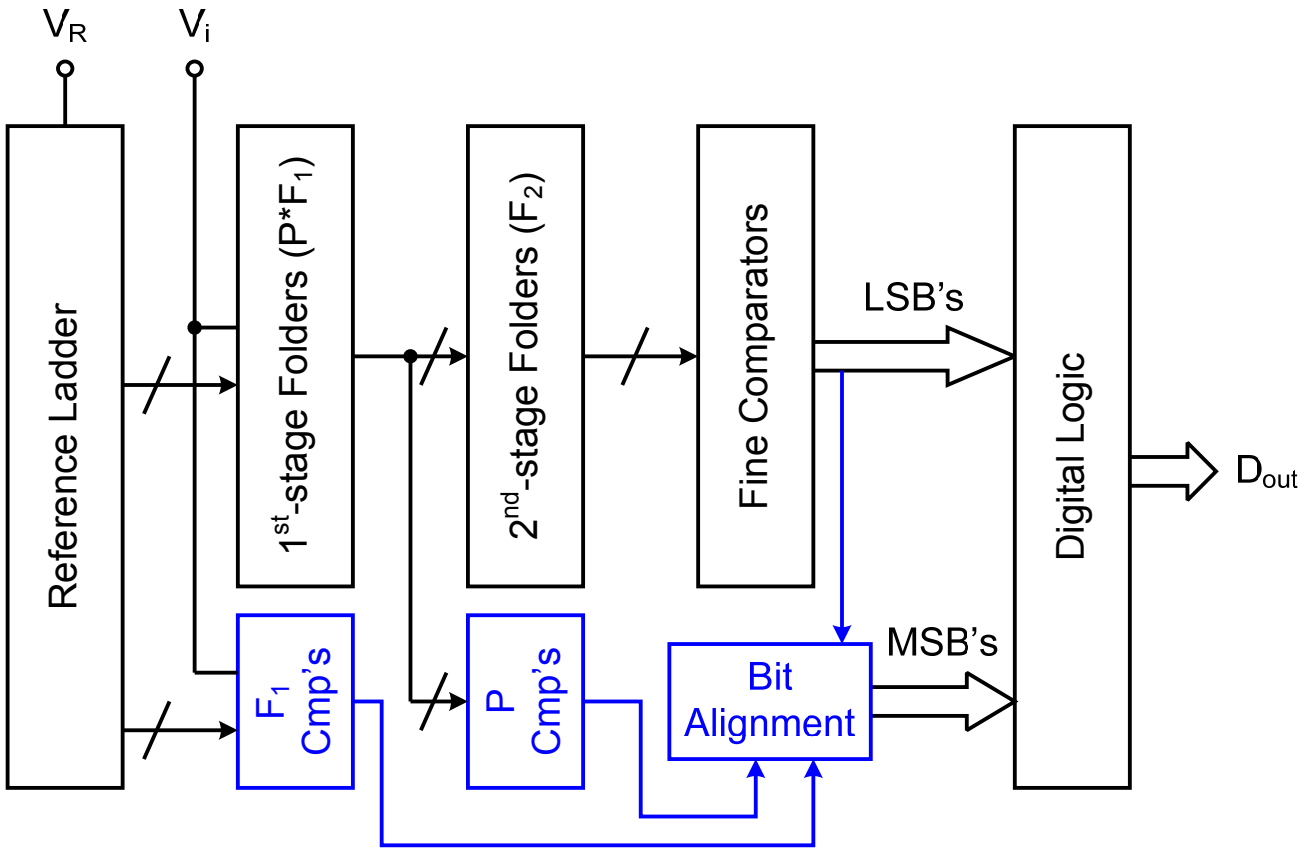
total # of coarse comparators = F_1*F_2, F_1+F_2, or F_1+P?

Cascaded Offset Bit Alignment (I)



- F_1 coarse comparators at input and P coarse comparators at 1st-stage folder outputs resolve F_1*P (>F_1*F_2) folds.
- One fine comparator output is utilized to perform offset bit alignment.

Cascaded Offset Bit Alignment (II)



Two-step offset bit alignment – large offset tolerance on F_1 coarse comparators and medium tolerance on P comparators.

References

1. Rudy van de Plassche, “CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters,” 2nd Ed., Springer, 2005..
2. Y. Chiu, *Data Converters Lecture Slides,* UT Dallas 2012.
3. B. Boser, *Analog-Digital Interface Circuits Lecture Slides,* UC Berkeley 2011.