E-Books και links σχετικά με ύλη μαθήματος

- *The Data Conversion Handbook*, by Walt Kester (*Newnes*, 2005)

http://www.analog.com/library/analogDialogue/archives/39-06/data_conversion_handbook.html

  – Ο συγγραφέας είναι senior engineer στην Analog Devices

Τα περιεχόμενα του βιβλίου και ειδικότερα το Chapter 3 ”Data Converter Architectures”, καλύπτει το μεγαλύτερο τμήμα του μαθήματος πλην των Folded και interpolated ADCs και των Sigma-Delta Converters

Για Sigma-Delta A/D Converters

- ADC Architectures III: Sigma-Delta ADC Basics by Walt Kester : MT-022 TUTORIAL –ANALOG DEVICES http://www.analog.com/media/en/training-seminars/tutorials/MT-022.pdf?doc=cn0354.pdf
- ADC Architectures IV: Sigma-Delta ADC Advanced Concepts and Applications by Walt Kester : MT-022 TUTORIAL – ANALOG DEVICES http://www.analog.com/media/en/training-seminars/tutorials/MT-023.pdf

E-Books και links σχετικά με ύλη μαθήματος

Για folded kai interpolation ADCs έχει χρησιμοποιηθεί input από ορισμένα PhD Theses τα παρακάτω:

- Κατά βάση τα Κεφάλαια 2 και 3 καθώς και την ενοτητα 4.3.1 από την MSc Thesis (MIT 2001):

A High-Speed Cascaded Folding and Interpolating A/D Converter

by

Yanlok Charlotte Lau

- Κεφάλαια 2.6 και Κεφ.3 από την PhD Thesis:

  DESIGN OF HIGH SPEED FOLDING AND INTERPOLATING ANALOG-TO-DIGITAL CONVERTER by YUNCHU LI (Texas A&M University 2003)

- Τα Κεφάλαια 1 και 2 από την Msc Thesis (περιληπτική αναφορά σε folded +interpolating ADCs):

  DESIGN OF A CMOS 6-BIT FOLDING AND INTERPOLATING ANALOG TO DIGITAL CONVERTER by Song Liu, University of Idaho, 1999

- Μπορείτε να τα βρείτε αυτά απλώς βάζοντας στο google search τους παραπάνω τίτλους ή/και από Νημερτή

E-Books και links σχετικά με ύλη μαθήματος

- Τα Κεφάλαια 2.6 και Κεφ.3 από την PhD Thesis:

  DESIGN OF HIGH SPEED FOLDING AND INTERPOLATING ANALOG-TO-DIGITAL CONVERTER by YUNCHU LI (Texas A&M University 2003)

- Τα Κεφάλαια 1 και 2 από την Msc Thesis (περιληπτική αναφορά σε folded +interpolating ADCs):

  DESIGN OF A CMOS 6-BIT FOLDING AND INTERPOLATING ANALOG TO DIGITAL CONVERTER by Song Liu, University of Idaho, 1999

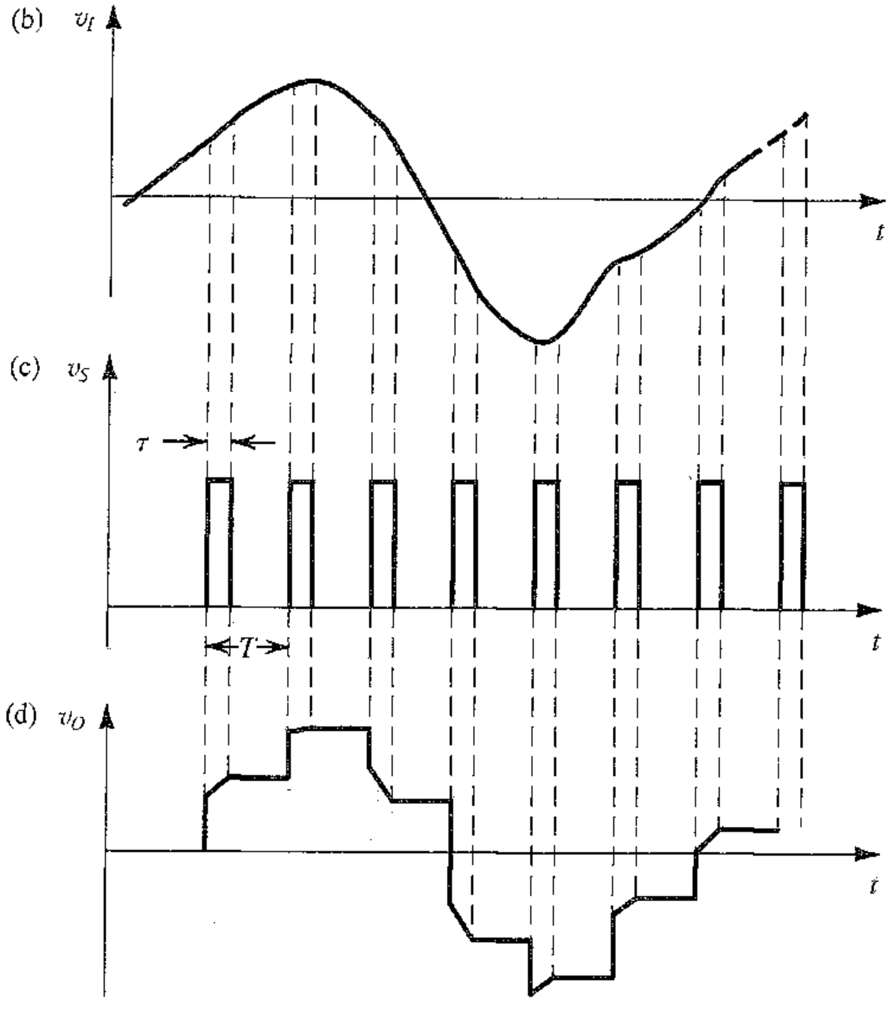
Μπορείτε να τα βρείτε αυτά απλώς βάζοντας στο google search τους παραπάνω τίτλους ή/και από Νημερτή

Με αυτά (και με καποιες μικρέςαναφορες που έχω κάνει από Sedra Smith και Richard Jaeger) είστε πλήρως καλυμμένοι από άποψη ύλης

Data Converters

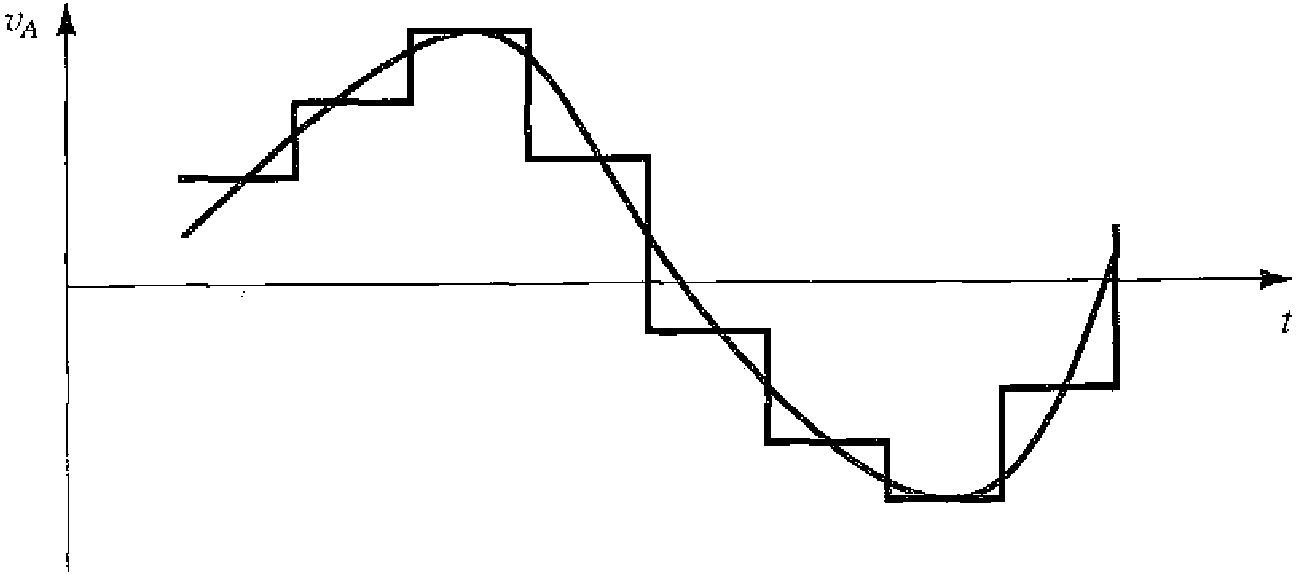
- Αν και το συντριπτικό ποσοστό του electronic processing λαμβάνει χώρα πλέον στον ψηφιακό κόσμο εν τούτοις ο πραγματικός κόσμος = αναλογικός=> Χρειάζεται:
  - Αναλογικό σε Ψηφιακό μετατροπέα (A/D converter -ADC） για εισαγωγή δεδομένων στο digital domain
  - Ψηφιακός σε Αναλογικό Μετατροπέα (D/A converter –DAC) για εξαγωγή αποτελεσμάτων στον πραγματικό αναλογικό κόσμο

Sampling



(***(b) Input signal waveform,*** ***(c) Sampling*** **signal** (control signal for the switch), **(d) Output signal** (to be fed to A/D converter or drive the output of the D/A).

Sampling (Signal Quantization)



The analog samples at the output of a D/A converter are usually fed to a sample-and-hold circuit to obtain the staircase waveform. This waveform can then be filtered to obtain the smooth waveform, shown.

It is evident that sampling rate should satisfy Nyquist criterion in order to be able to reconstruct the original (analog) waveform

-Usually 1.5 to 10 times the Nyquist rate.

Sampling (Signal Quantization)

Consider an analog signal whose values range from 0 to +10 V. We wish to convert it to a 4-bit digital signal.

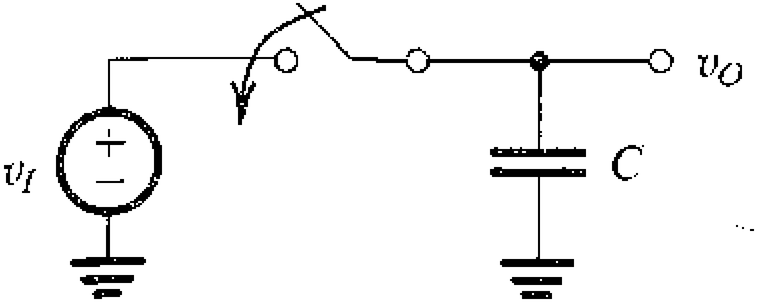
- A 4-bit binary number can represent 16 different values, 0 to 15; -The *resolution of the conversion will be 10 V / 1 5 = 2/3 V.*
- *Thus an analog signal of 0 V will be* represented by 0000, 2\3 V will be represented by 0001, 6 V will be represented by 1001, and 10 V will be represented by 1111.
- All these sample numbers are multiples of the basic increment (2 /3 V).

A question now arises :What if the conversion of numbers fall between these successive incremental levels.?

- For instance, consider the case of a 6.2-V analog level. This falls between 18/3 and 20/3 .However, since it is closer to 18/3 we treat it as if it were 6 V and *code it as 1001. This* process is called **quantization.**
- **Obviously errors are inherent in this process; such errors are** called quantization errors.
- Using more bits to represent (encode or, simply, code) an analog signal reduces quantization errors but requires more complex circuitry.
- The quantization error of an A /D converter is equivalent to ±1 least significant bit

Sampling Circuit

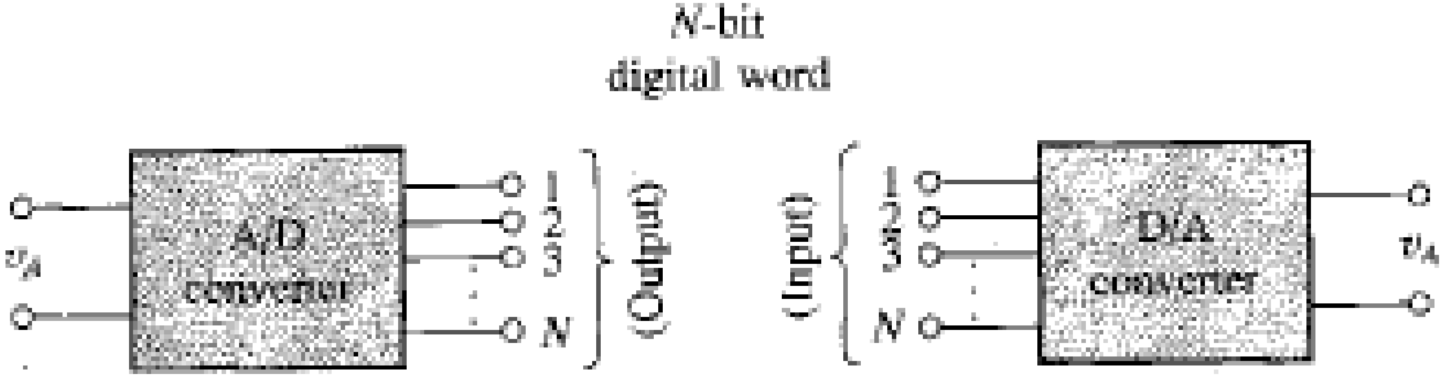
- Τόσο ως είσοδο στον A/D όσο και ως έξοδο του D/A απαραίτητη= η δειγματοληψία των αναλογικών σημάτων, παρακάτω απεικονίζεται ένα τετοιο τυπικό κύκλωμα σε μορφή blockδιαγράμματος



a) Sample-and-hold (S/H) circuit. The switch closes for a small part (T seconds) of every clock period ***(T).***

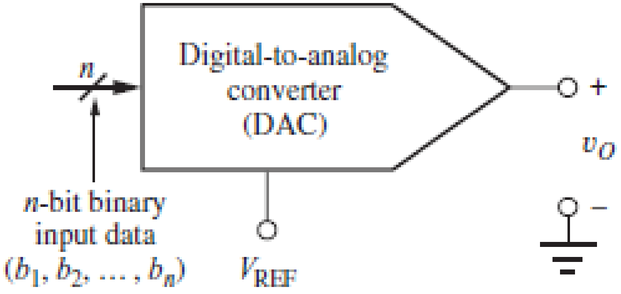
Sample –and – Hold Circuit: Αποτελείται από έναν αναλογικό διακόπτη (transmission gate) , πυκνωτή αποθήκευσης και buffer amplifier (μη ορατός στο σχήμα)

The A/D and D/A Converters as Functional Blocks



DACs find numerous applications, from trimming and calibration circuits to high-end video DACs, and communication circuits.

D/A CONVERTER FUNDAMENTALS



Assuming a voltage output, the behavior of the DAC can be expressed as

$$v_O = V_{FS}(b_1 2^{-1} + b_2 2^{-2} + \cdots + b_n 2^{-n}) + V_{OS}$$

The DAC output may also be a current that can be represented as

$$i_O = I_{FS}(b_1 2^{-1} + b_2 2^{-2} + \cdots + b_n 2^{-n}) + I_{OS}$$

The **full-scale voltage** ***VFS or full-scale current IFS is related to the internal reference voltage VREF*** of the converter by

$$V_{FS} = K V_{\mathrm{REF}} \qquad \text{or} \qquad I_{FS} = G V_{\mathrm{REF}}$$

K and G determine the gain of the converter

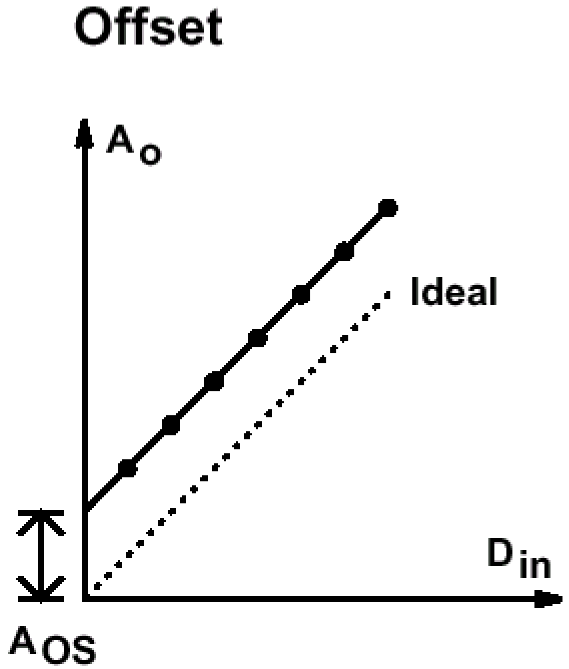
VOS and IOS represent the ***offset voltage or offset current of the converters,*** *and* characterize the converter output when the digital input code is equal to zero

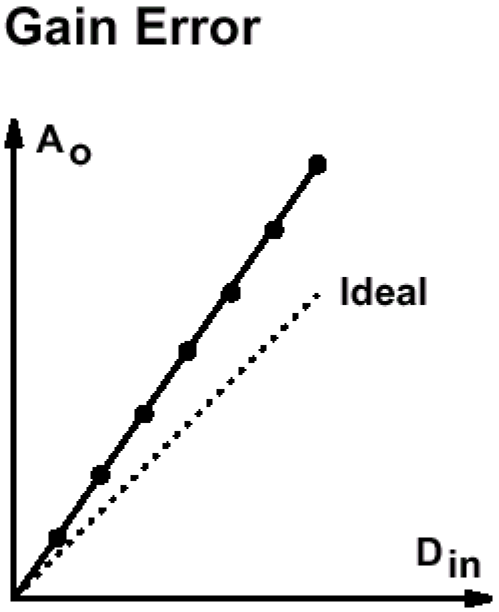
The smallest voltage change that can occur at the DAC output takes place when the **least significant bit (LSB)** ***bn in the digital word changes from a 0 to a 1. This minimum voltage change*** is also referred to as the **resolution of the converter and is given by** $V_{\mathrm{LSB}} = 2^{-n} V_{FS}$

Characteristics of ADC and DAC

- DAC
  - Monotonic and nonmonotonic
  - Offset , gain error , DNL and INL
  - Glitch
  - Sampling-time uncertainty
- ADC
  - missing code
  - Offset , gain error , DNL and INL
  - Quantization Noise
  - Sampling-time uncertainty

D/A Converter Errors (Offset and Gain Errors)

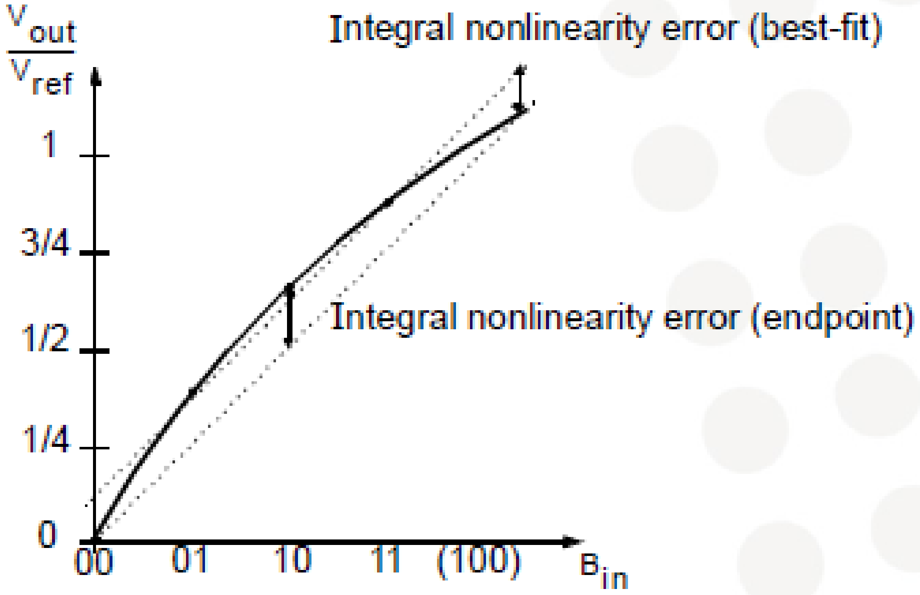

Offset


Gain Error

In a D/A converter (”DAC”) the ***offset error*** *is defined to be the* output that occurs for the input code that should provide zero output

The **gain error** is the difference at the full scale value between ideal and actual curves when the offset error has been reduced to zero. For a DAC it is given in units of LSBs.

Integral nonlinearity error (INL)



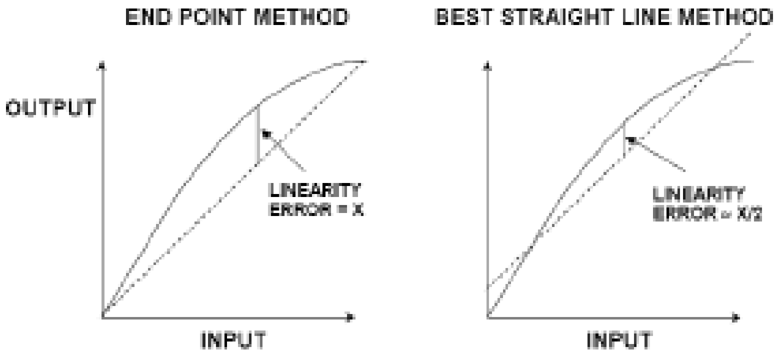
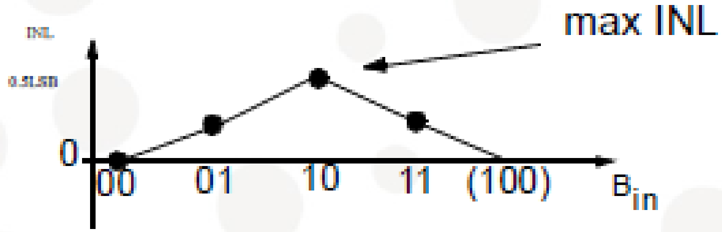
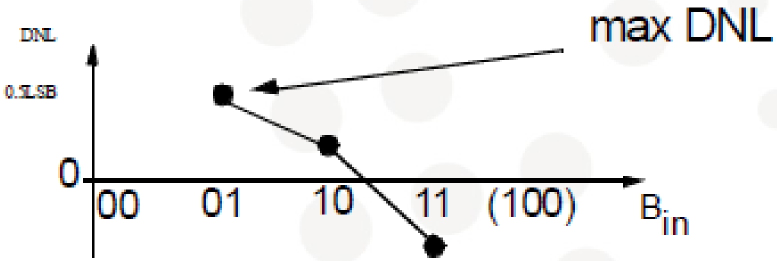


Figure 2.18: Method of Measuring Integral Linearity Errors (Same Converter on Both Graphs)



After both offset and gain errors have been removed, the ntegral nonlinearity (INL) error is defined to be the deviation from a straight line. Possible straight lines: endpoints of the converters transfer respons, best-fit straight line such tha the difference (or mean squared error) is minimized.

Differential nonlinearity error (DNL)



step size between 00 and 01 is 1.5 LSB
step size between 10 and 11 is 0.7 LSB

•Ideally, each analog step size is equal to 1 LSB. DNL is variation in step size from V_{LSB} (after removal of gain and offset errors). Ideally DNL is 0 for all digital values. DNL is in "J & M" defined for each digital word, whereas other sometimes refer to DNL as the maximum magnitude of DNL values.

D/A CONVERTER ERRORS (Example)

Figure 12.21 and columns 1 and 2 in Table 12.7 present the relationship between the digital input code and the analog output voltage for an ideal three-bit DAC.

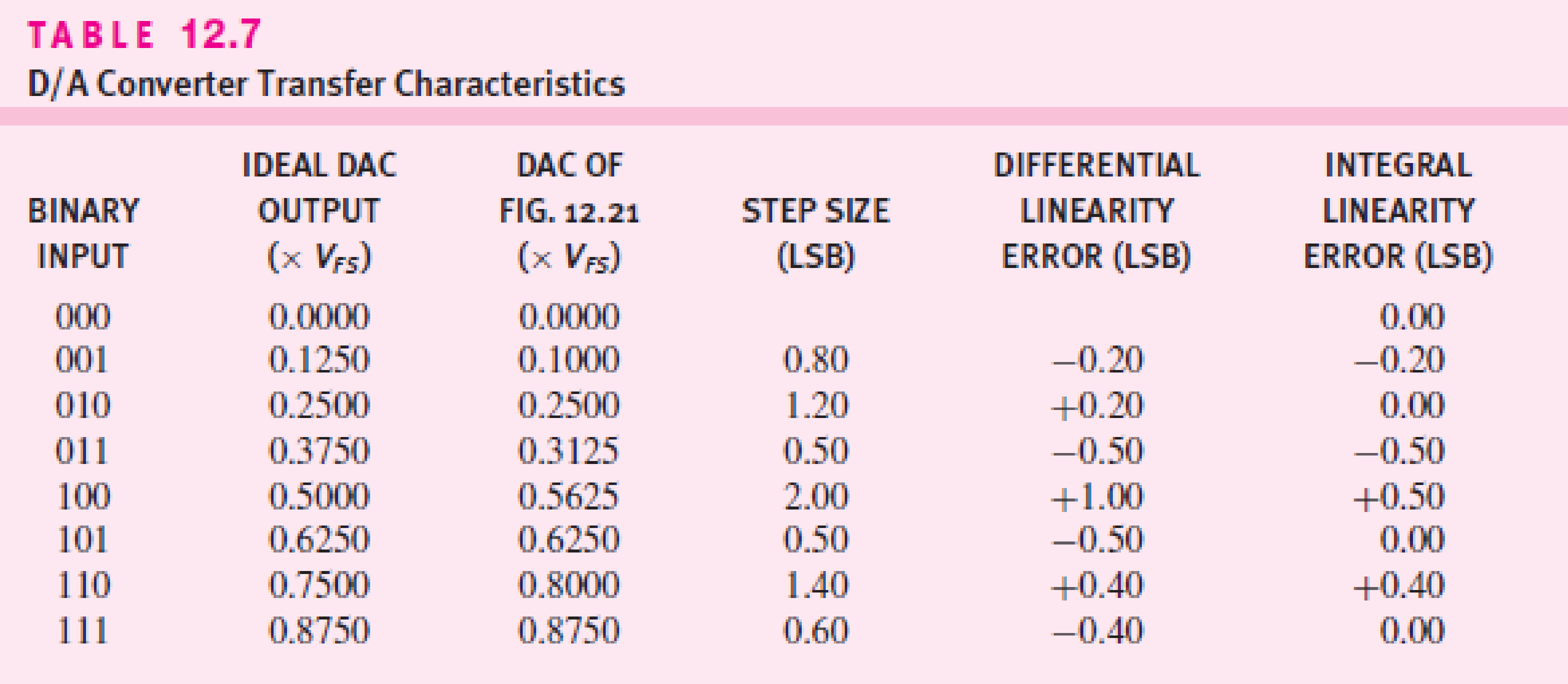


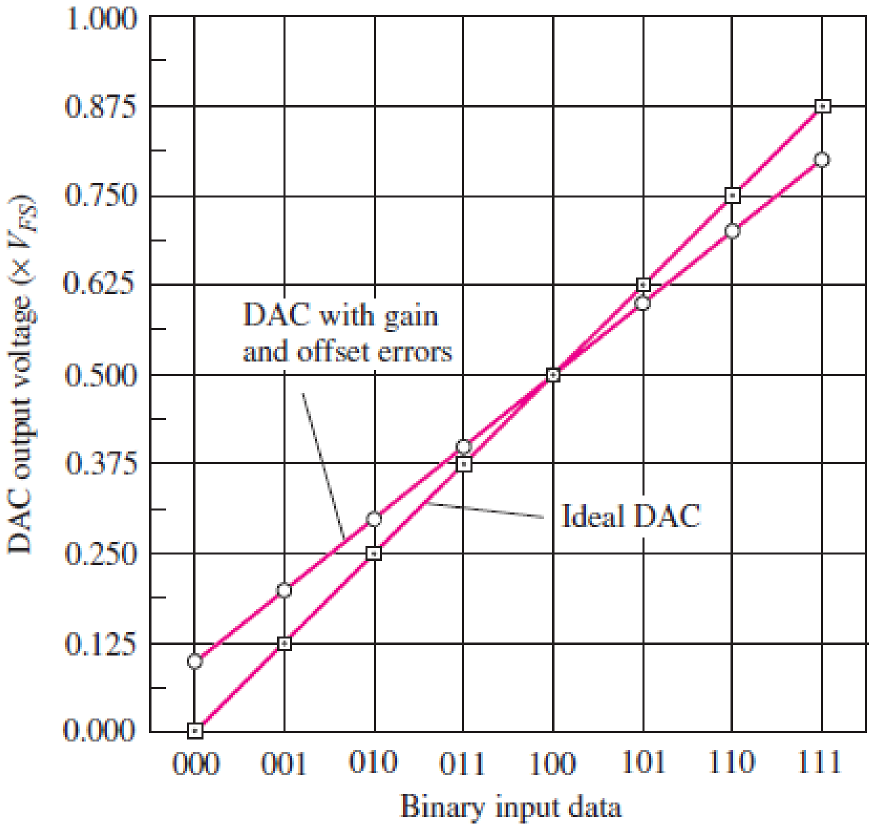
TABLE 12.7

D/A Converter Transfer Characteristics

BINARY INPUT	IDEAL DAC OUTPUT ($\times V_{FS}$)	DAC OF FIG. 12.21 ($\times V_{FS}$)	STEP SIZE (LSB)	DIFFERENTIAL LINEARITY ERROR (LSB)	INTEGRAL LINEARITY ERROR (LSB)
000	0.0000	0.0000			0.00
001	0.1250	0.1000	0.80	−0.20	−0.20
010	0.2500	0.2500	1.20	+0.20	0.00
011	0.3750	0.3125	0.50	−0.50	−0.50
100	0.5000	0.5625	2.00	+1.00	+0.50
101	0.6250	0.6250	0.50	−0.50	0.00
110	0.7500	0.8000	1.40	+0.40	+0.40
111	0.8750	0.8750	0.60	−0.40	0.00

The data points in the figure represent the eight possible output voltages, which range from 0 to 0.*875 × VFS. Note that the output voltage* of the ideal DAC never reaches a value equal to *VFS. The maximum output is always 1 LSB smaller* than *VFS. In this case, the maximum output code of 111 corresponds to 7/8 of full scale or 0.875 VFS*

D/A CONVERTER ERRORS (Example)



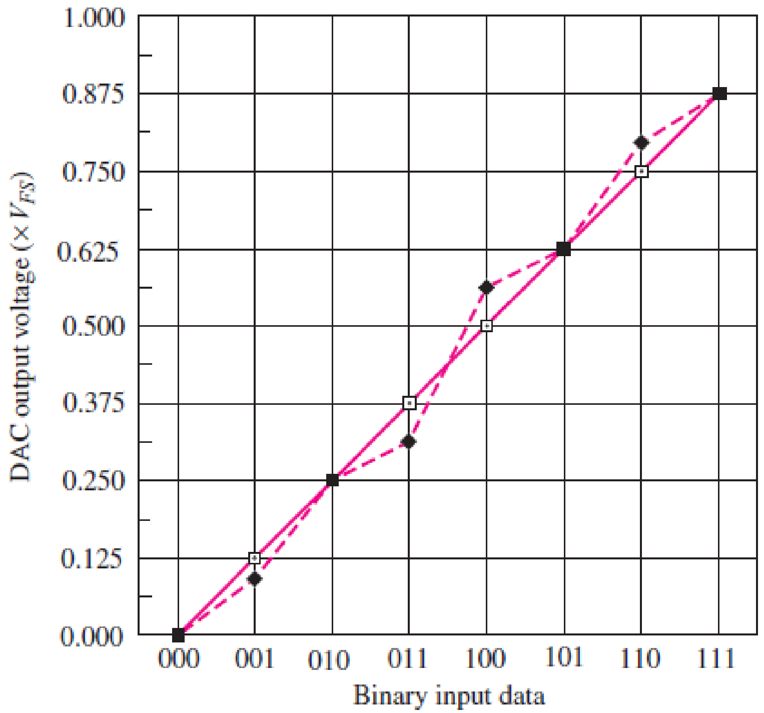
Transfer characteristic for an ideal DAC and a converter with both gain and offset errors.

The ideal converter in above Fig has been calibrated so that *VOS = 0 and 1 LSB =VFS/8. Figure 1.21 also shows the output of a converter with both gain and offset errors.*

The ***gain*** **error of the D/A converter represents the deviation of the slope of the converter transfer function** from that of the corresponding ideal DAC (previous Figure)

The offset voltage is simply the output of the converter for a zero binary input code

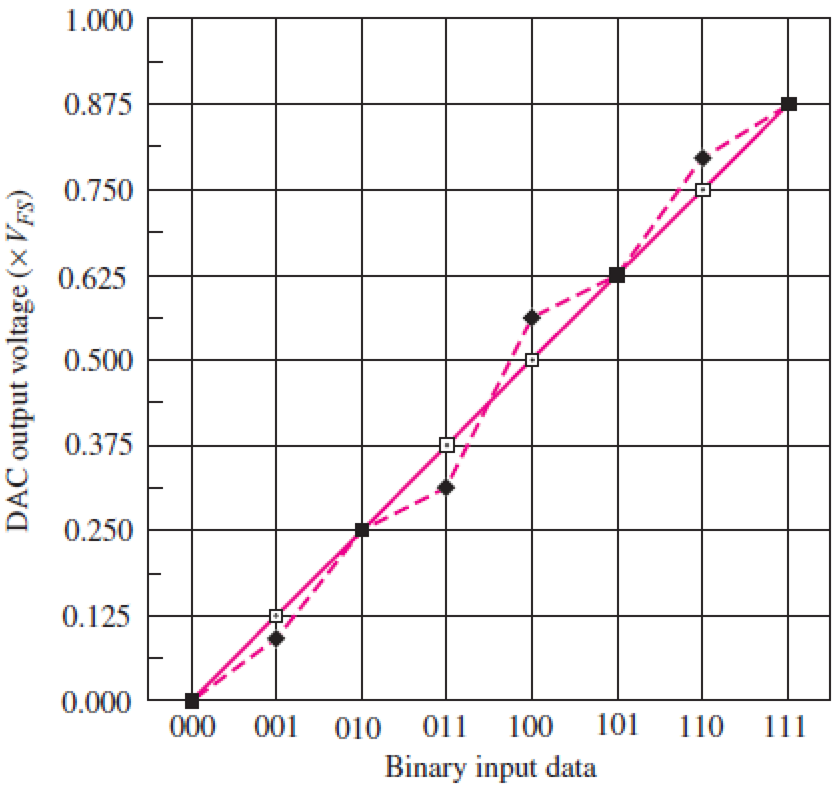
D/A CONVERTER ERRORS (Example)



The converter contains circuit mismatches that cause the output to no longer be perfectly linear. **Integral linearity error, or just linearity error, measures the deviation of the actual converter output** from a straight line fitted to the converter output voltages. The error is specified as a fraction of an LSB or as a percentage of the full-scale voltage.

Linearity errors for inputs 001, 011, 100, and 110. The overall linearity error for the DAC is specified as the magnitude of the largest error that occurs. Hence this converter will be specified as having a linearity error of either 0.5 LSB or 6.25 percent of full-scale voltage. A good converter exhibits a linearity error of less than 0.5 LSB.

D/A Converter Errors (Example



A closely related measure of converter performance is the **differential linearity error. When the** binary input changes by 1 bit, the output voltage should change by 1 LSB.

A converter's differential linearity error is the magnitude of the maximum difference between each output step of the converter and the ideal step size of 1 LSB.

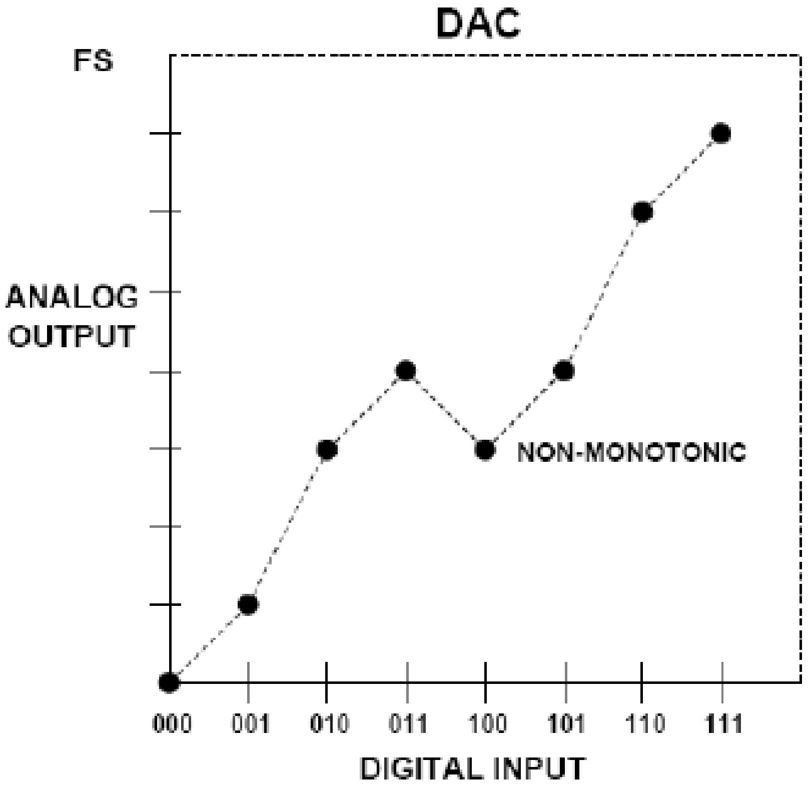
For instance, (Table 2.7) DAC output changes by 0.8 LSB when input changes from 000 to 001.

The differential linearity error represents the difference between this actual step size and 1 LSB.

The integral linearity error for a given binary input represents the sum of the differential linearity errors for inputs up through the given input.

Monotonicity in DACs

- A monotonic DAC is one in which the output always increases as the input increases (slope of the transfer response is of only one sign.)
- If the maximum DNL error is less than 1 LSB, the DAC is guaranteed to be monotonic.
- However, many monotonic converters may have a maximum DNL greater than 1 LSB.
- Similarly, a converter is guaranteed to be monotonic if maximum DNL is < 1 LSB.
- 3-bit nonmonotonic example in the figure is from Analog-Digital conversion handbook by Analog Devices



D/A Techniques

1. D/A Converters using Binary-Weighted Elements

Binary Weighted Resistors

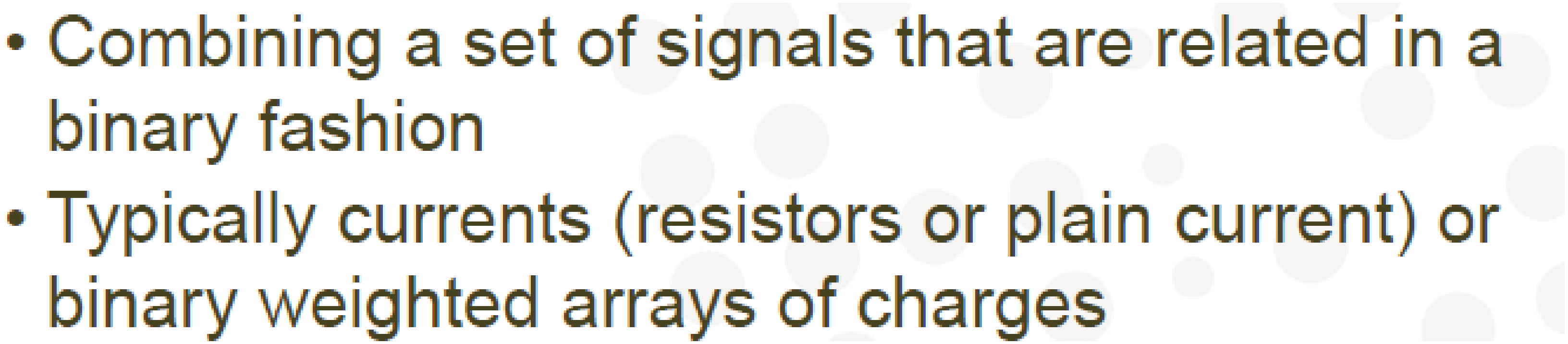
R-2-R Ladders

Binary Weighted Capacitors

Binary Weighted current sources

A/D Converters using Binary-Weighted Elements

General Concept

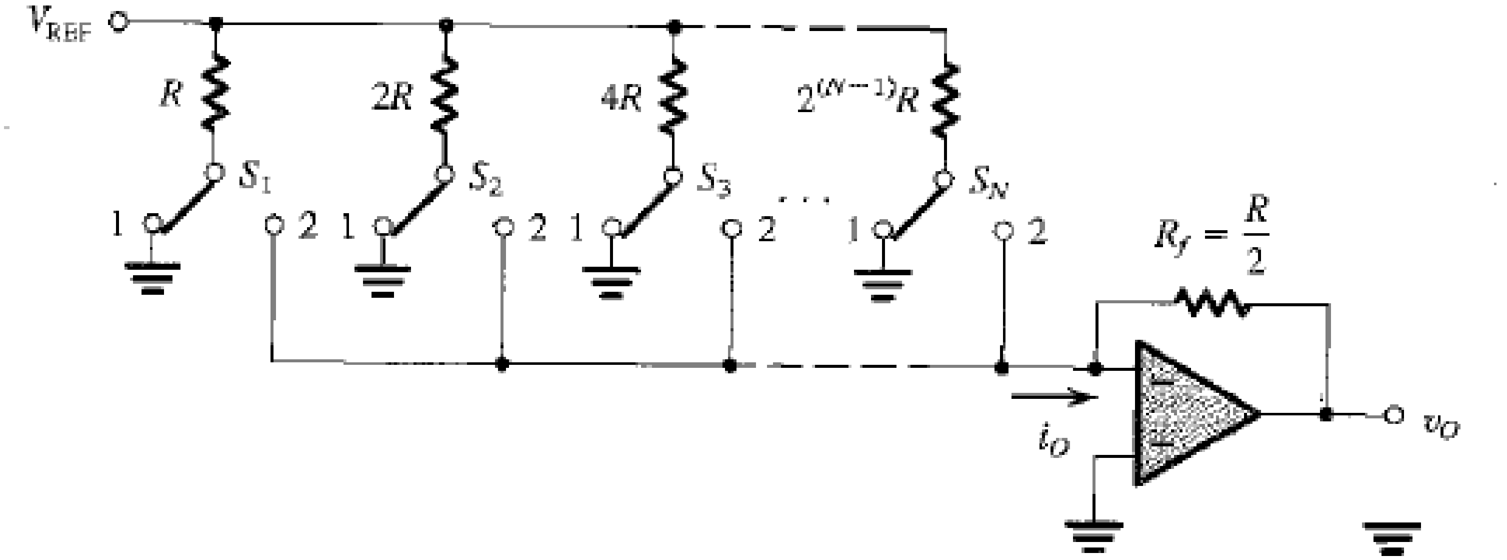


- Combining a set of signals that are related in a binary fashion
- Typically currents (resistors or plain current) or binary weighted arrays of charges

Binary Weighted Resistors

The circuit consists of :

A reference voltage VREF, N binary-weighted resistors *R, 2R, 4R, 8R, . . . , 2exp(N-1)R, N* switches *S1,2, • • . , SN, and an op amp together with its feedback resistance*



$$i_O = \frac{V_{REF}}{R}b_1 + \frac{V_{REF}}{2R}b_2 + \cdots + \frac{V_{REF}}{2^{N-1}R}b_N$$

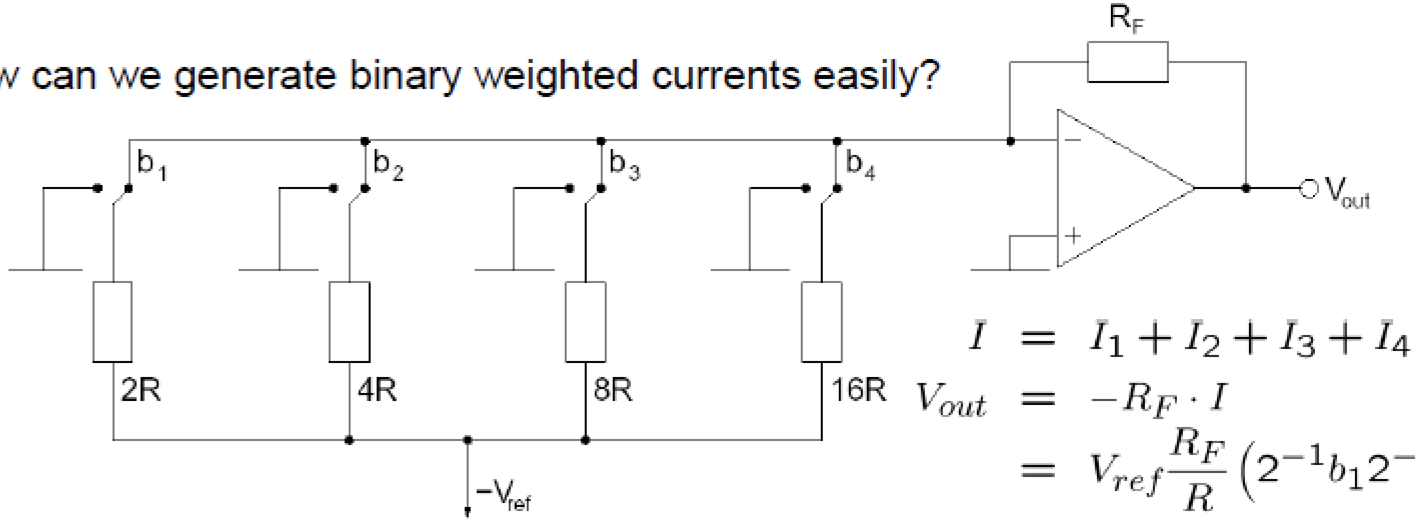
$$= \frac{2V_{REF}}{R}\left(\frac{b_1}{2^1} + \frac{b_2}{2^2} + \cdots + \frac{b_N}{2^N}\right)$$

$$D = \frac{b_1}{2^1} + \frac{b_2}{2^2} + \cdots + \frac{b_N}{2^N}$$

Binary Weighted Resistors (4-bit case)

Implementation of Binary Weighted DAC

How can we generate binary weighted currents easily?



$$
\begin{aligned}
I &= I_1 + I_2 + I_3 + I_4 \\
V_{out} &= -R_F \cdot I \\
&= V_{ref}\frac{R_F}{R}\left(2^{-1}b_1 2^{-2}b_2 + \right. \\
&\qquad \left. + 2^{-3}b_3 + 2^{-4}b_4\right)
\end{aligned}
$$

$$
\begin{aligned}
I_1' &= -V_{ref} \cdot \frac{1}{2R} & \qquad I_1 &= -V_{ref} \cdot \frac{1}{2R} \cdot b_1 \\
I_2' &= -V_{ref} \cdot \frac{1}{4R} & \qquad I_2 &= -V_{ref} \cdot \frac{1}{4R} \cdot b_2 \\
I_3' &= -V_{ref} \cdot \frac{1}{8R} & \qquad I_3 &= -V_{ref} \cdot \frac{1}{8R} \cdot b_3 \\
I_4' &= -V_{ref} \cdot \frac{1}{16R} & \qquad I_4 &= -V_{ref} \cdot \frac{1}{16R} \cdot b_4
\end{aligned}
$$

Binary Weighted Resistors

Discussion: Advantages –Disadvantages

- Popular for bipolar technology.
- Few switches and resistors-Number of resistors = N ≪ 2expN as holds in other cases
- Large resistance ratios = 2expN.
- Scaled switches for large current ratios.
- No guarantee of monotonicity (holds for most case of A/Ds based on binary scaled elements)
- Prone to Glitches in high-speed operation., if switches do not change simultaneously

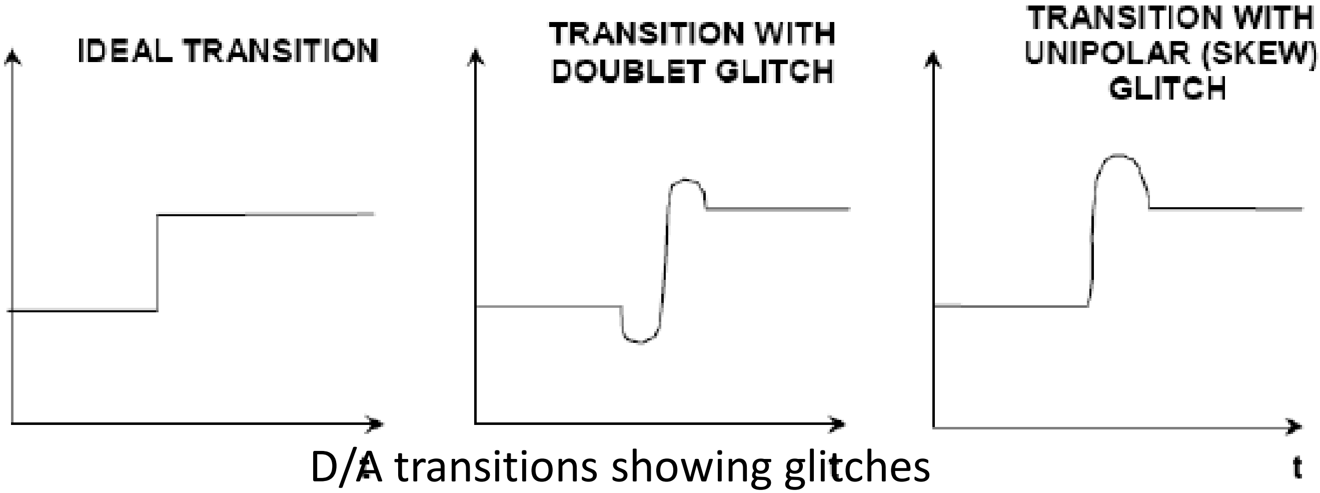
Conclusion: Not practical for many bits (N>4)

A more convenient scheme exists utilizing a resistive network caled R-2R ladder

Glitches

(from Analog Digital Conversion Handbook)

Ideally, when a DAC output changes it should move from one value to its new one monotonically. In practice, the output is likely to overshoot, undershoot, or both (see Figure 2.94). This uncontrolled movement of the DAC output during a transition is known as a *glitch*. It can arise from two mechanisms: capacitive coupling of digital transitions to the analog output, and the effects of some switches in the DAC operating more quickly than others and producing temporary spurious outputs.


IDEAL TRANSITION
TRANSITION WITH DOUBLET GLITCH
TRANSITION WITH UNIPOLAR (SKEW) GLITCH
t
t
t

D/A transitions showing glitches

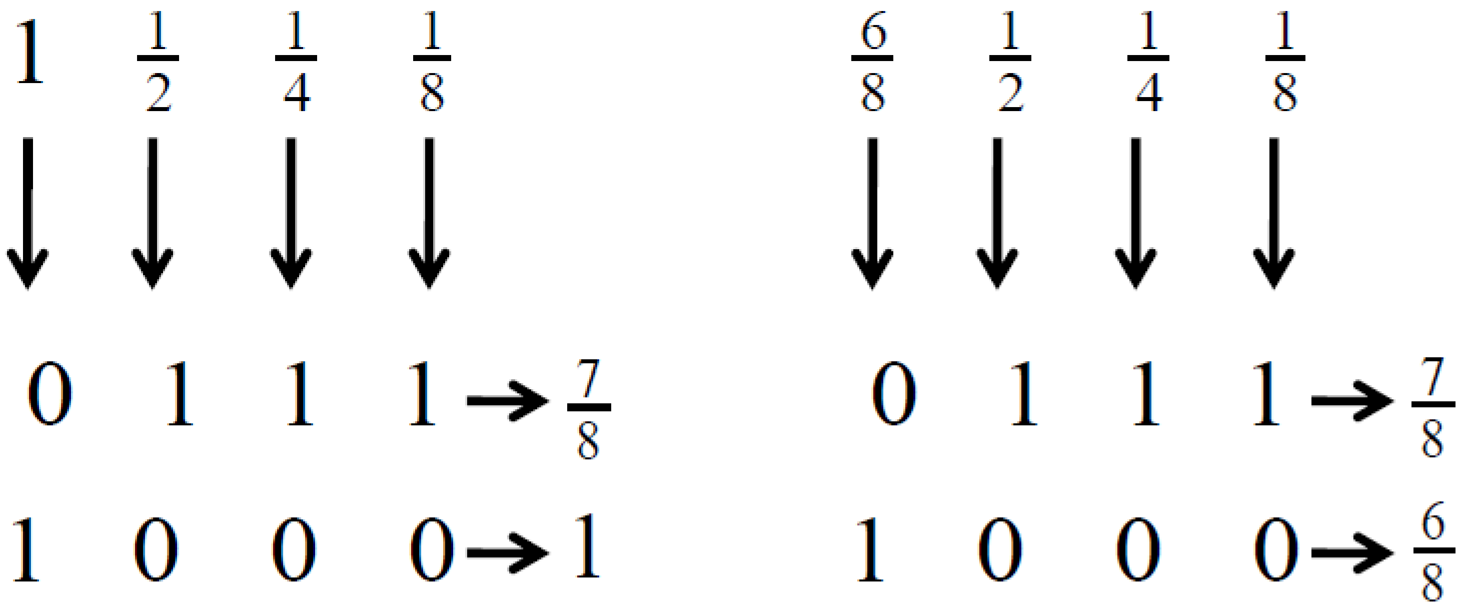
Glitches waste energy and make noise

Potential cures:

- Exact matching in time (difficult)
- Add S/H to the output

Monotonicity in Binary Weighted DACs

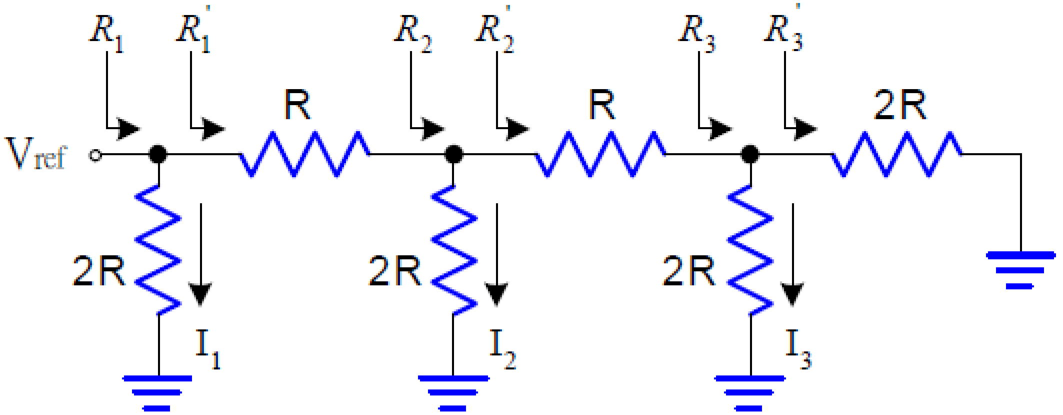
- Binary weighted converters are not necessarily monotonic
- Example:



$$\begin{array}{cccccc} 1 & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & & \\ \downarrow & \downarrow & \downarrow & \downarrow & & \\ 0 & 1 & 1 & 1 & \rightarrow & \frac{7}{8} \\ 1 & 0 & 0 & 0 & \rightarrow & 1 \end{array} \qquad \begin{array}{cccccc} \frac{6}{8} & \frac{1}{2} & \frac{1}{4} & \frac{1}{8} & & \\ \downarrow & \downarrow & \downarrow & \downarrow & & \\ 0 & 1 & 1 & 1 & \rightarrow & \frac{7}{8} \\ 1 & 0 & 0 & 0 & \rightarrow & \frac{6}{8} \end{array}$$

R-2R Ladder

Basic Idea



$$R_3' = 2R$$

$$R_3 = 2R // R_3' = R$$

$$R_2' = R + R_3 = 2R$$

$$R_2 = 2R // R_2' = R$$

=> $I_1 = \frac{V_{ref}}{2R}$

Also, the voltage at node 2 is one-half the voltage at node 1, giving

$$I_2 = \frac{V_{ref}}{4R}$$

At node 3, the voltage divides in half once again, therefore

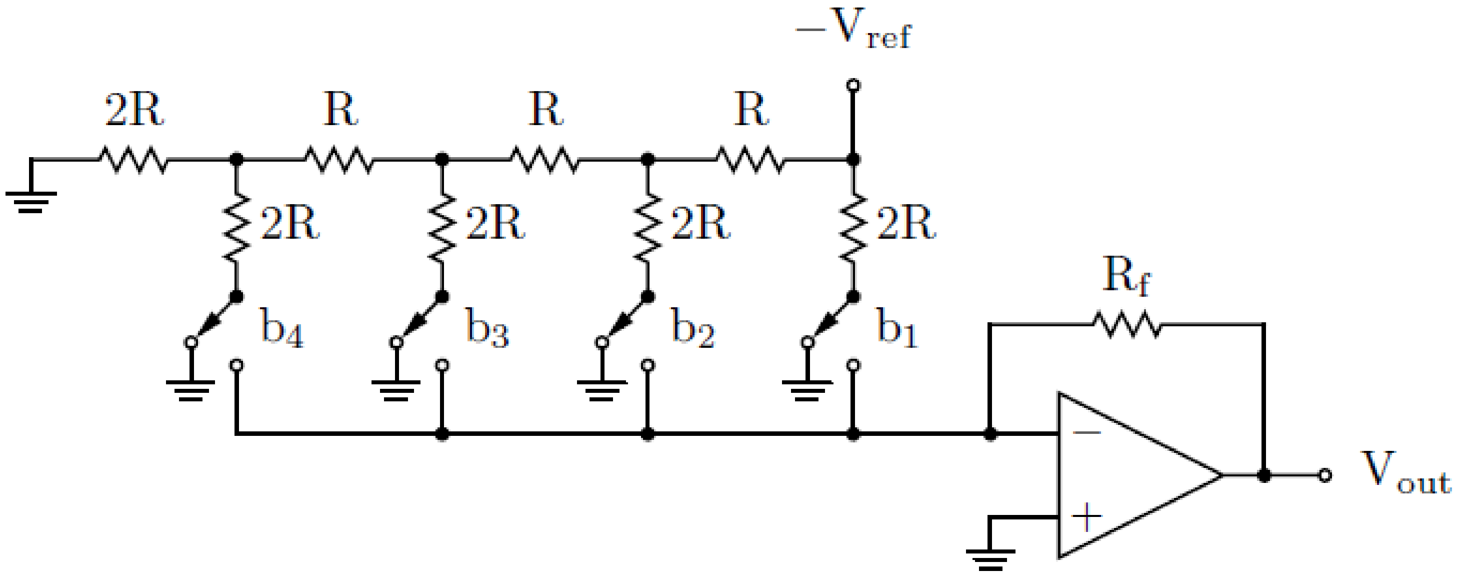
$$I_3 = \frac{V_{ref}}{8R}$$

and so on. Thus the R-2R ladder can obtain binary-weighted currents

R-2R Ladder

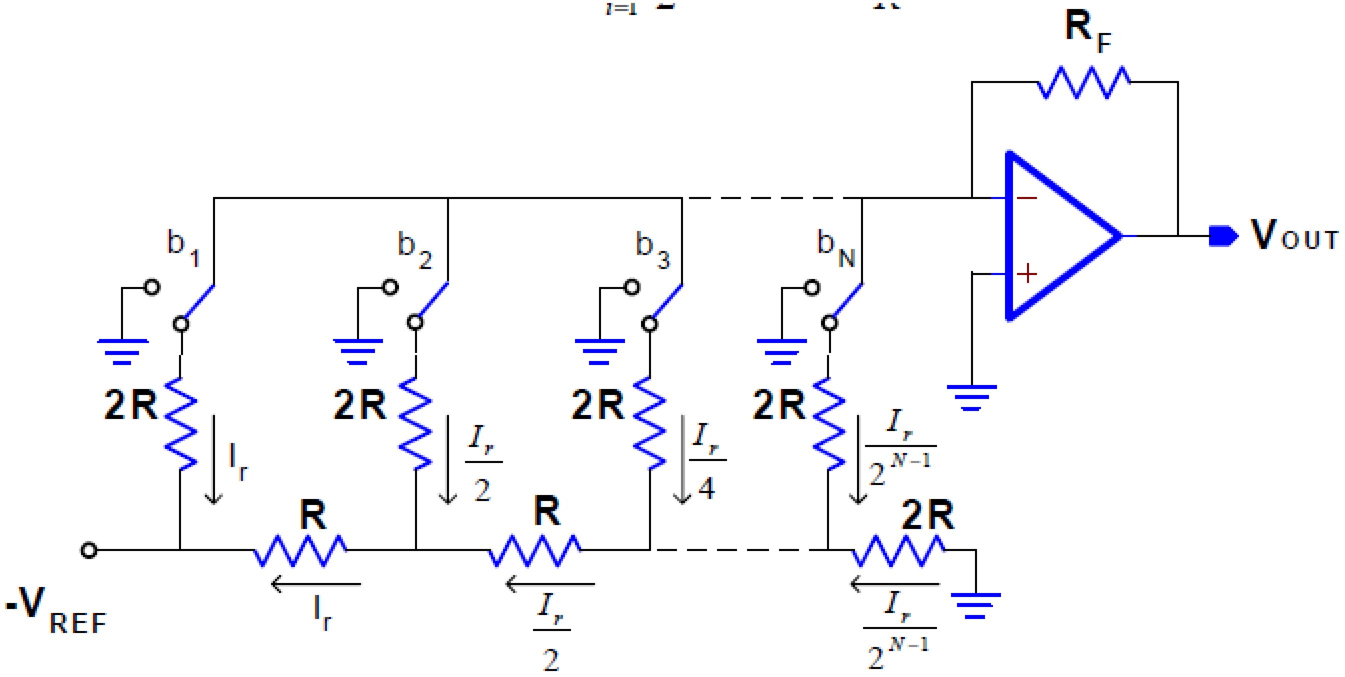
Current-driven converter: inverted R-2R ladder.

$$V_{out} = -R_f \frac{V_{ref}}{R} \left(-\frac{b_1}{2} - \frac{b_2}{4} - \frac{b_3}{8} - \frac{b_4}{16} \right) = \frac{R_f}{R} V_{ref} B_{in}$$



R-2R Ladder

Current-driven converter: inverted R-2R ladder-N bits



$$I_r = V_{REF}/2R$$

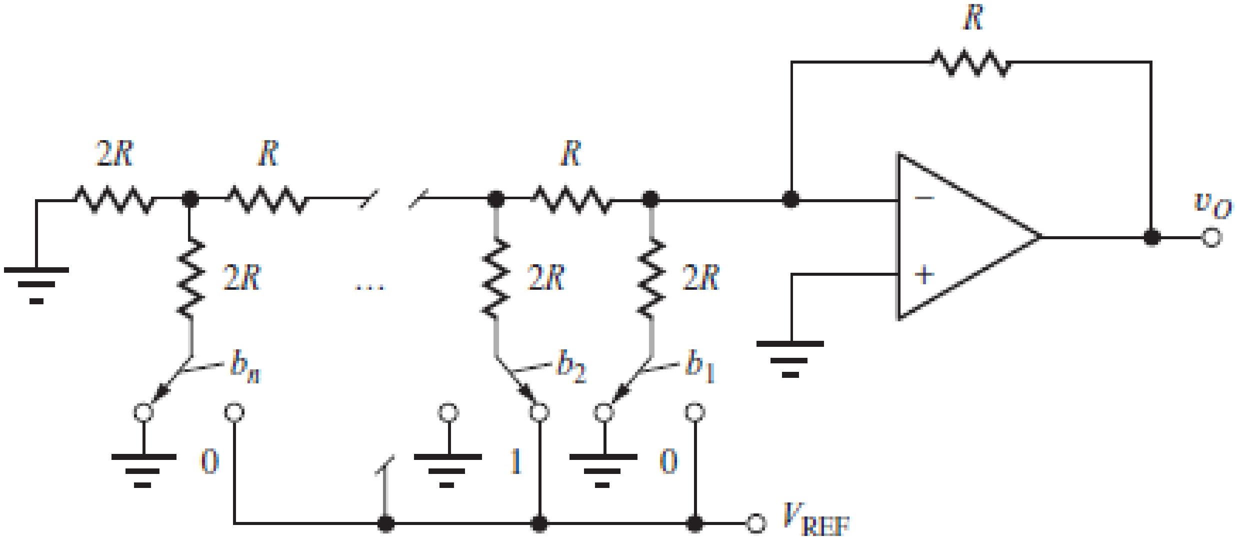
$$V_{OUT} = R_F \sum_{i=1}^{N} \frac{b_i I_r}{2^{i-1}} = V_{REF}\left(\frac{R_F}{R}\right) B_{in}$$

Important to scale the switches accordingly

- Ensuring equal voltage drop across the switches

R-2R Ladder

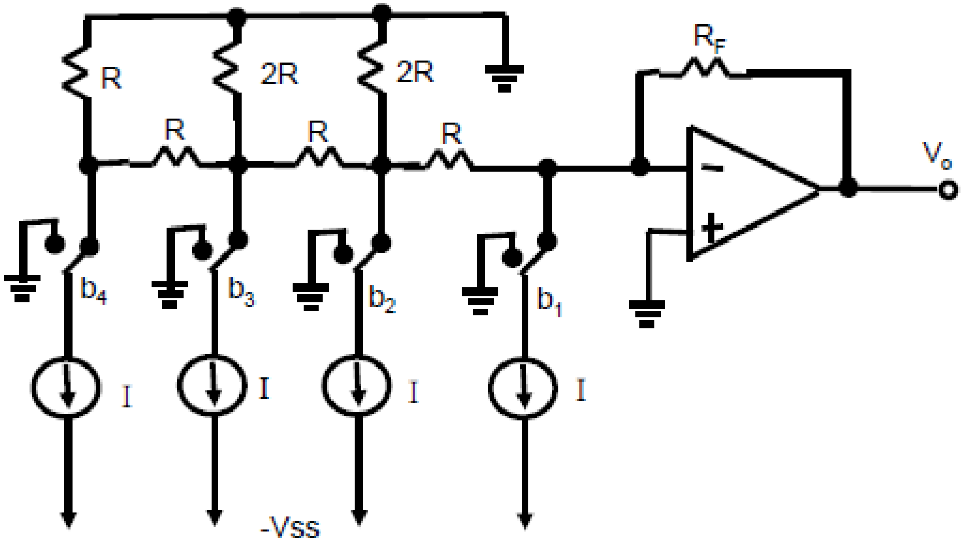
Voltage-driven converter: 4-bit R-2R based DAC.



R-2R Ladder

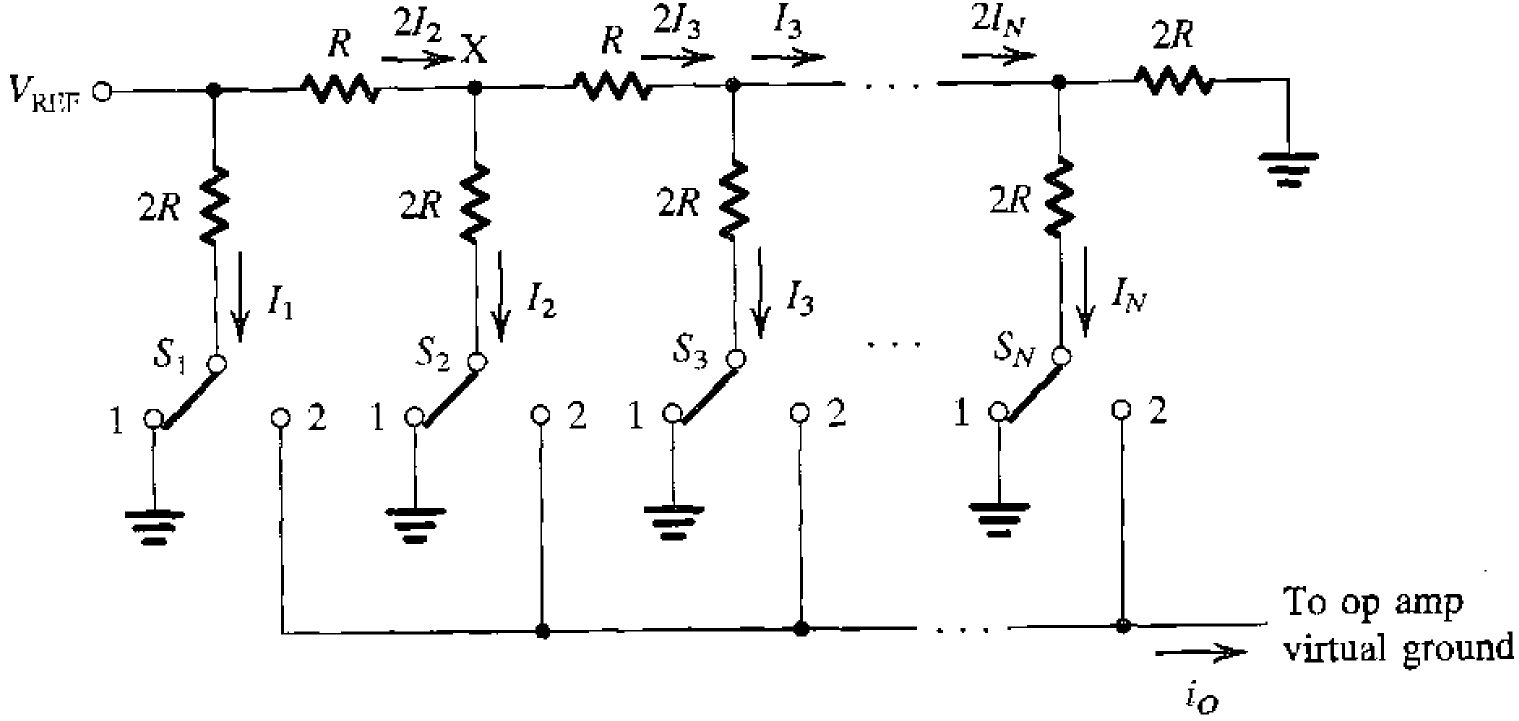
- Discussion: Advantages –Disadvantages
- Very popular architecture.
- Binary-weighted currents by R-2R ladder.
- Number of resistors = 2N ≪ 2expN.
- Small resistance ratio = 2 ≪ 2exp N: independent of N.
- R = 2 ∼ 10 k, 2R = R + R to improve matching.
- Still prone to glitches and no guarrantee that they are monotonic
- Current ratio is still large =>large ratio of switch sizes
  - Scaled switches for large current ratios: 1/2 ∼ 1/(2expN).
- Faster and satisfactory performance until 8 bits

R-2R-Based DAC
(driven by equal currents)



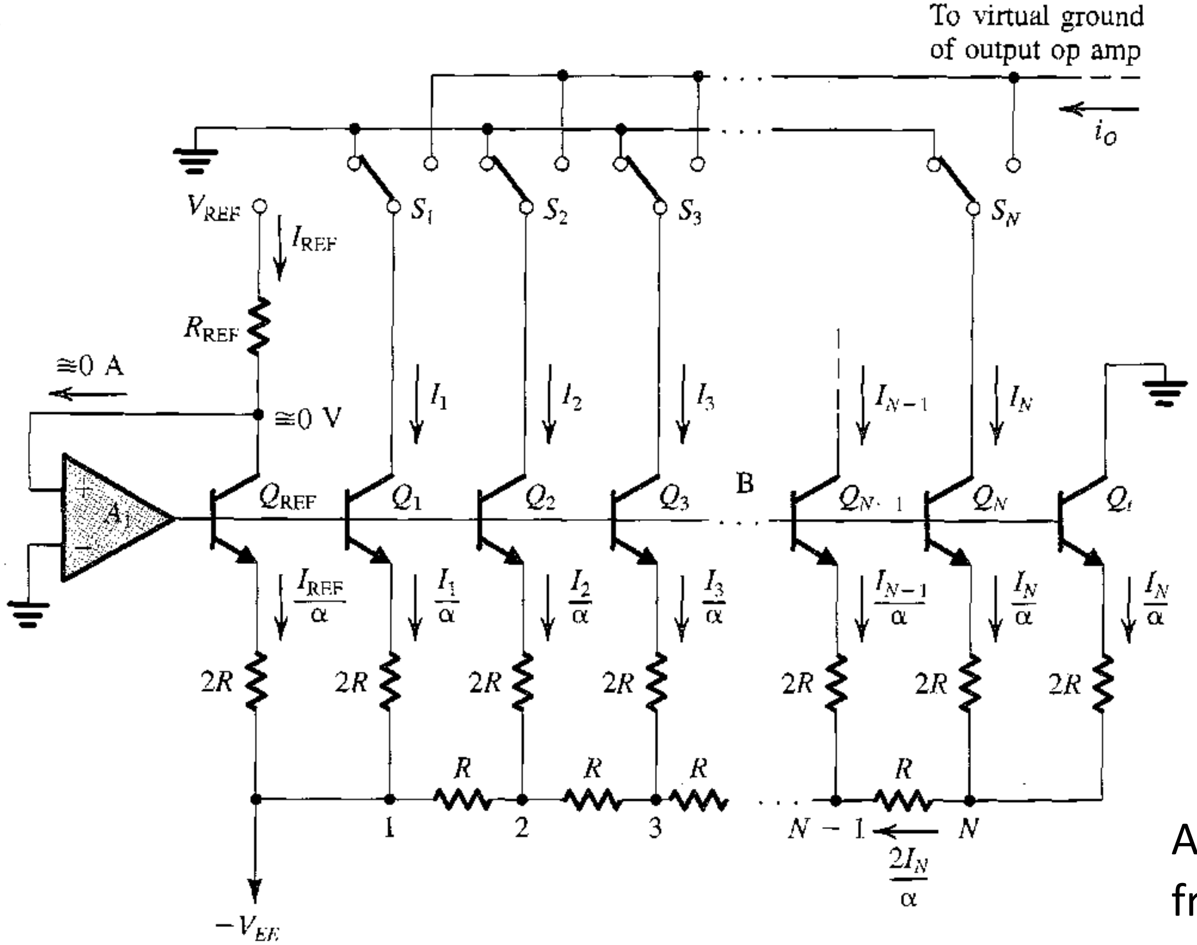
- R-2R ladder DAC driven by equal currents through switches- R-2R network performs the binary scaling of currents
- Slower since the internal nodes exhibit some voltage swings(as opposed to the previous configuration where internal nodes all remain at fixed voltage)
- Not necessary to scale switch sizes (Equal currents)

A Practical Circuit Implementation
(of a current driven R-2R Ladder DAC)



The R-2R Ladder A/D circuit the implementation of which will be described in the slides to follow

A Practical Circuit Implementation
(of a current driven R-2R Ladder DAC)



Aim is to show that I1 to IN= binary weighted

Considering Qn,Qt matched=> Ie for both= I_N/α

Thus voltage V_N between baseline and node N is :

$$V_N = V_{BE_N} + \left(\frac{I_N}{\alpha}\right)(2R)$$

and Voltage VN-1 between node B and N-1 will be:

$$V_{N-1} = V_N + \left(\frac{2I_N}{\alpha}\right)R = V_{BE_N} + \frac{4I_N}{\alpha}R$$

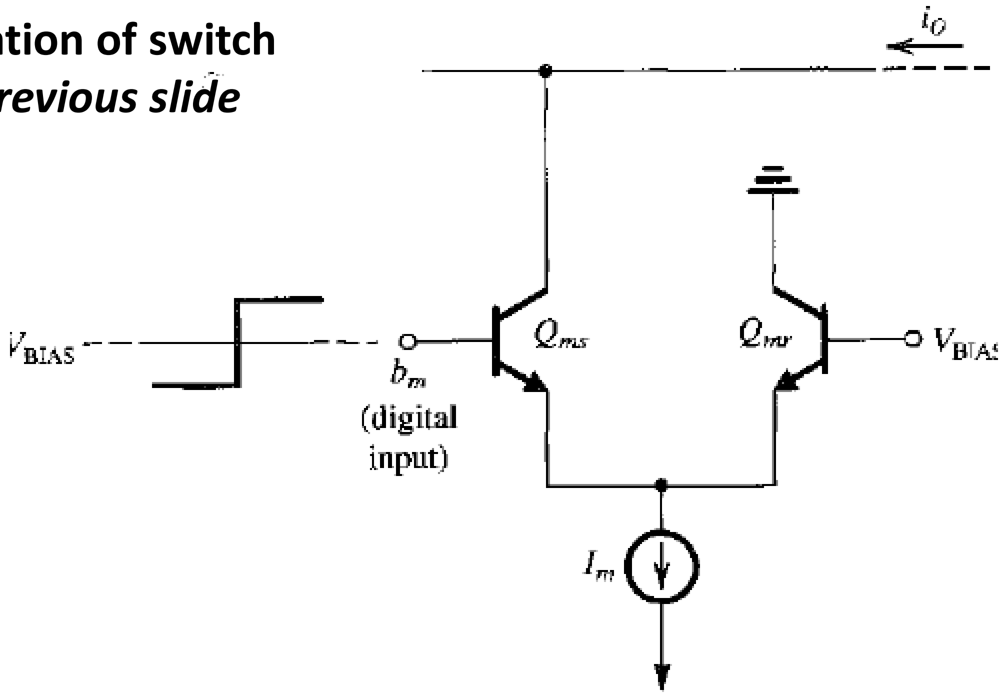
Assuming that V_{BE}(N-1)= V_{BE} (N), from above equation => Ie of QN-1 = 2IN/α=> Ie of QN-1=2 * Ie of QN Working in a similar way it can be shown that I1=2I2=4I3=…2exp(N-1) IN

Also notice that Collector of Qref is at virtual ground => its Ic= Vref/Rref . Considering Qref, Q1 are matched => their Ics will be equal too=> I1=Iref=>binary weighted currents =K * Iref independent of the values of VBE and α

A Practical Circuit Implementation

(of a current driven R-2R Ladder DAC)-Current Switches

Circuit implementation of switch
Sm in the DAC of previous slide



If V*bm is higher than* V_{BIAS} *by a few hundred millivolts, Qms will turn on and Qmr will turn off. The bit current Im will flow through Qms and onto the output summing line.* When *bm is low, Qms will be off and Im will flow through Qmr to ground.*

This current switch is simple and features high-speed operation. It suffers, however, from the fact that part of the current Im flows through the base of *Qms and thus does* not appear on the output summing line.

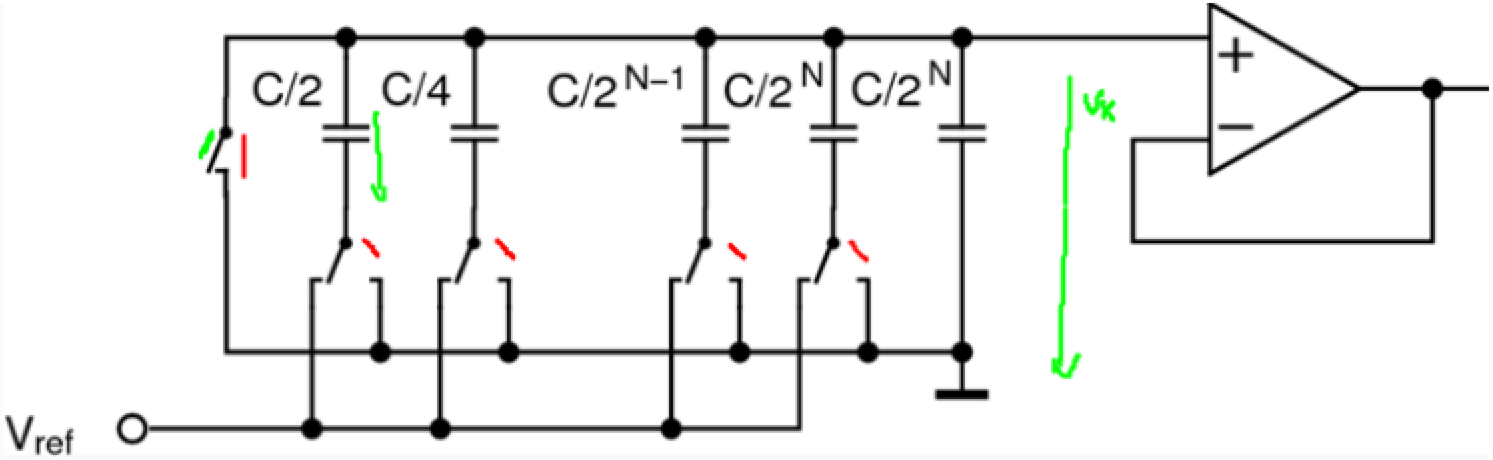
in a BiCMOS technology the differential-pair transistors *Qms and Qmr can be replaced with MOSFETs, thus eliminating the base current problem.*

Charge scaling DACs

Charge Scaling DACs operate by binarily dividing the total charge applied to a capacitor array

Two phased ϕ1 and ϕ2 non overlapping clocks-During ϕ1 capacitors are decharged

Normal operation during ϕ2 where they are connected either to Vref or ground depending on the value of the corresponding controlling bit



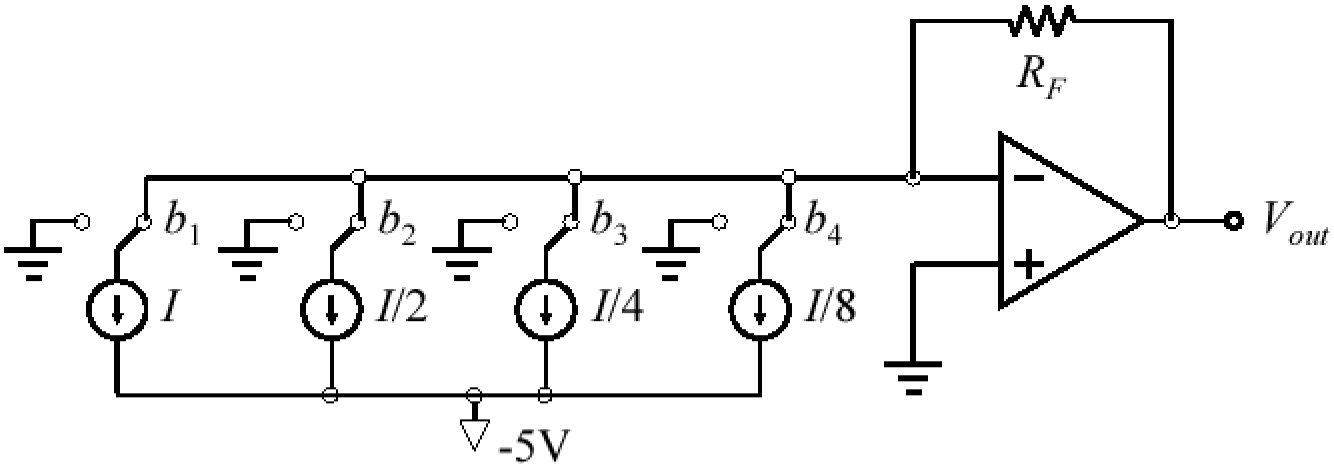
By equalizing the sum of charge Qi of each individual ci , to Ctotal x Vout and considering that Ctotal= Sum (ci) =C it is derived :

$$V_{out} = [b_1 2^{-1} + b_2 2^{-2} + \dots\dots + b_N 2^{-N}] \cdot V_{ref}$$

Matching =better than other (R-string) DACs -Accuracy and capacitor area=limiting factors

Big Problem =leakage –lose their accuracy in few msecs-suitable though for successive approximation ADcs

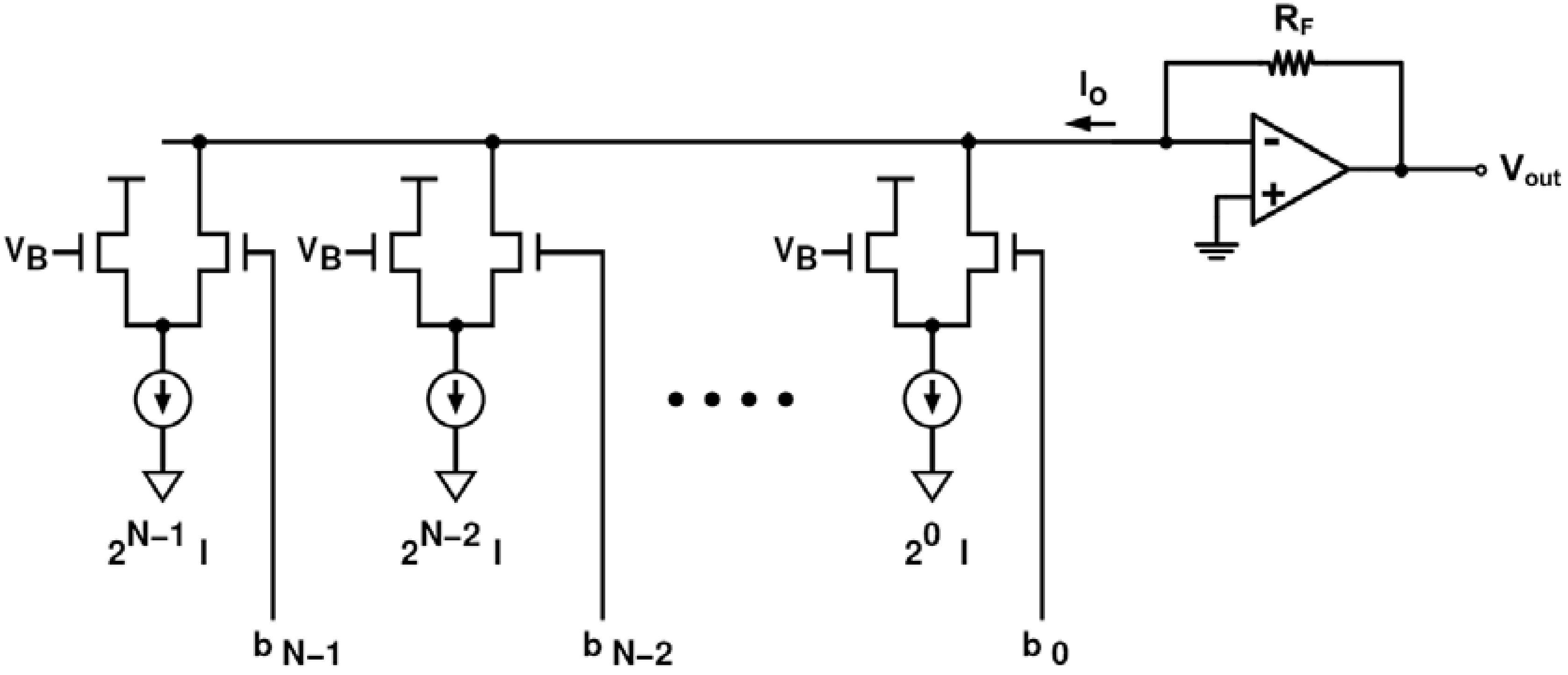
Current Mode D/As



$$I_o = I \times (b_{N-1} \cdot 2^{N-1} + b_{N-2} \cdot 2^{N-2} + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0)$$

- Current-mode DACs are very similar to resistor based converters, but intended for higher speed applications
- The basic idea is to switch currents to either the output or to ground,
- The output current is converted to a voltage through the use of the R_F resistance

Current Mode D/As (Current Source Detail)



A real example

(Analog Devices IC-1970)

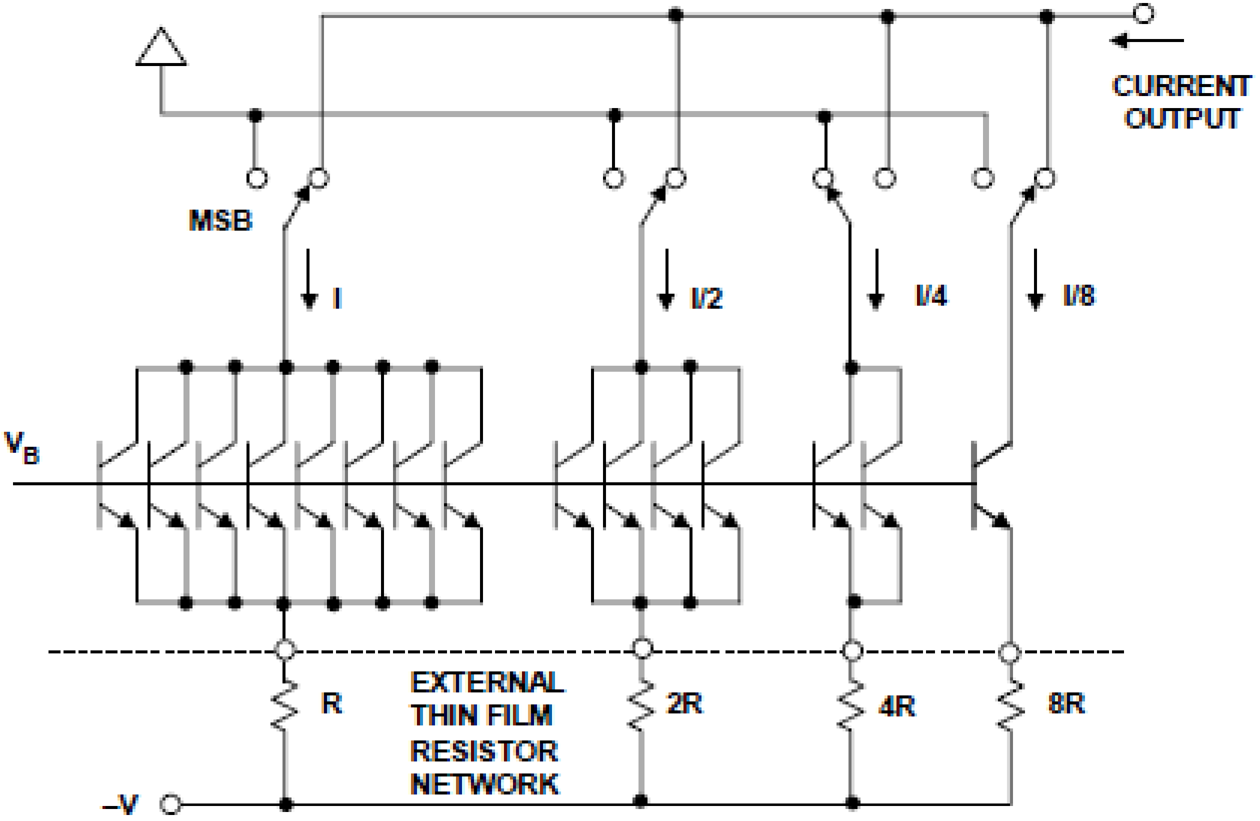
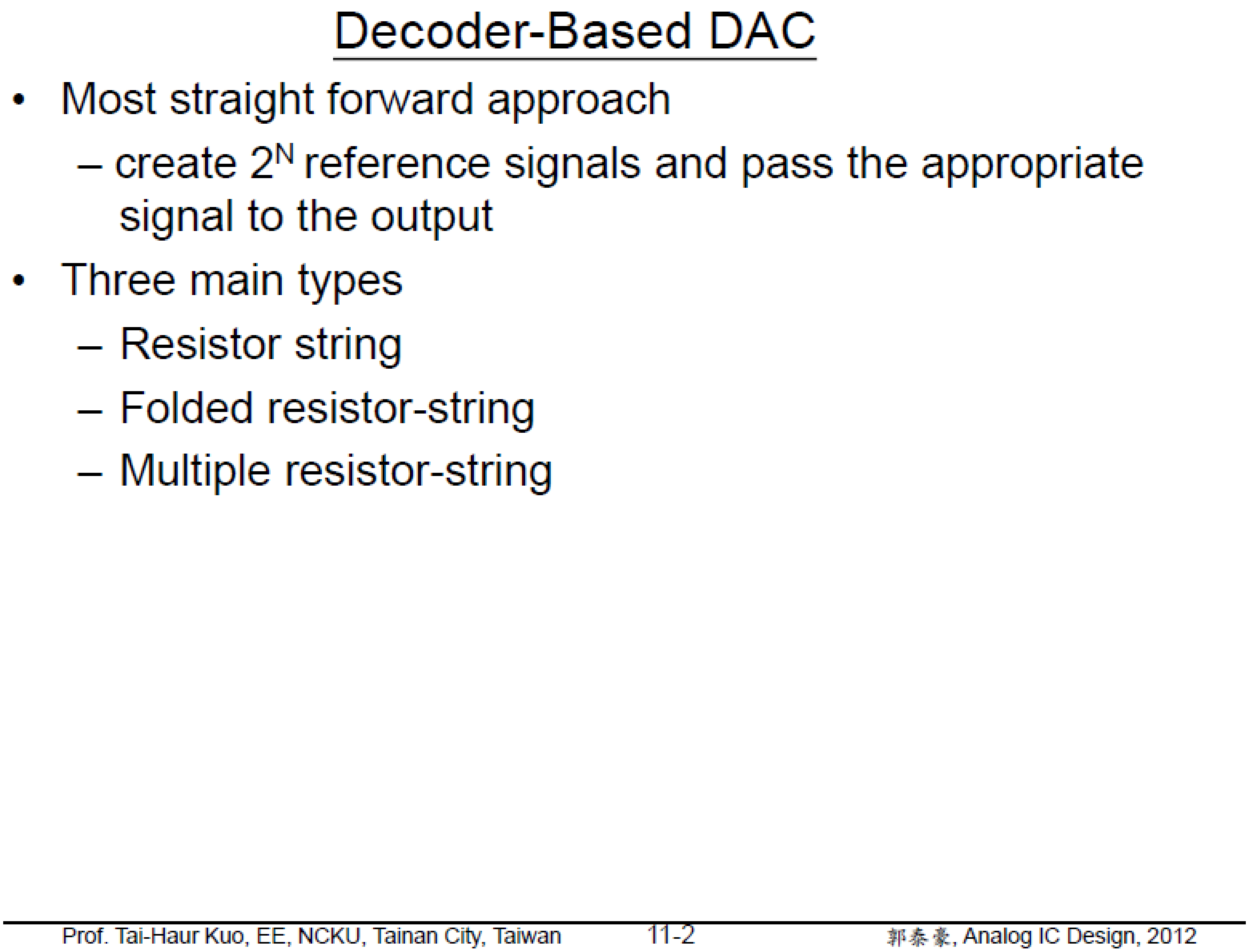


***Figure 3.20:** Binary-Weighted 4-Bit DAC, the AD550 "µDAC" Quad Switch*

R-String -Decoder-based DACs

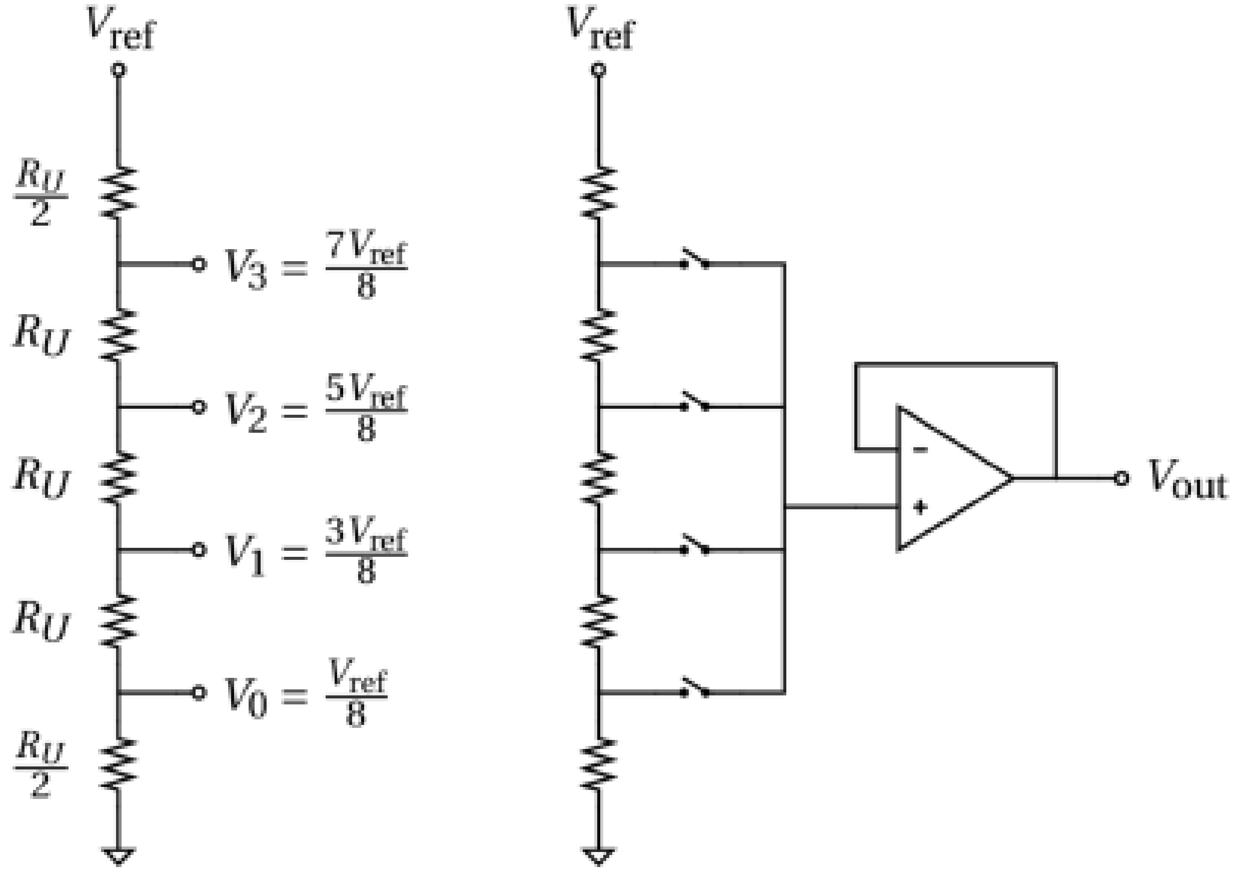


Decoder-Based DAC

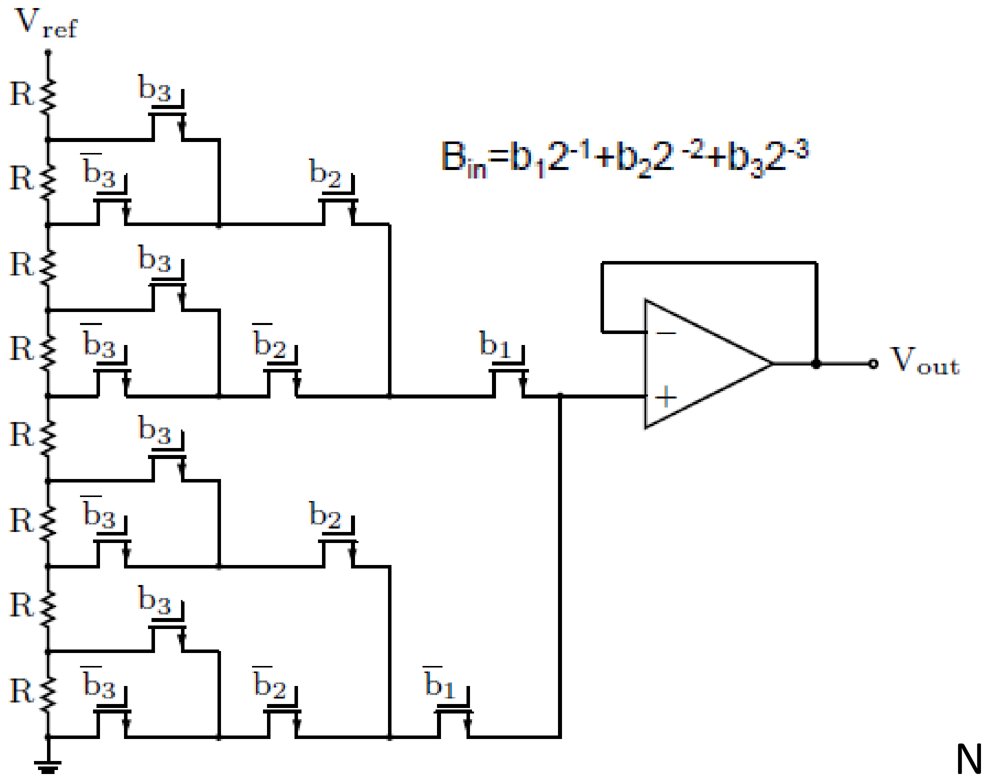
- Most straight forward approach
  - create 2^N reference signals and pass the appropriate signal to the output
- Three main types
  - Resistor string
  - Folded resistor-string
  - Multiple resistor-string

Resistor string converters

Basic Concept –No decoding Logic shown


V_{ref}
$\frac{R_U}{2}$
$V_3 = \frac{7V_{ref}}{8}$
R_U
$V_2 = \frac{5V_{ref}}{8}$
R_U
$V_1 = \frac{3V_{ref}}{8}$
R_U
$V_0 = \frac{V_{ref}}{8}$
$\frac{R_U}{2}$
V_{ref}
-
+
V_{out}

Resistor String Converter (with pass-gate tree decoder)



$B_{in}=b_1 2^{-1}+b_2 2^{-2}+b_3 2^{-3}$

- Switches = NMOS transistors
- Transmission gates enable
  - –higher voltage range
  - --but higher parasitic cap, area (layout more complicated)
- Buffer experiences high input voltage variation
- Slow due to buffer and analog mux
- How fast does the DAC settle?

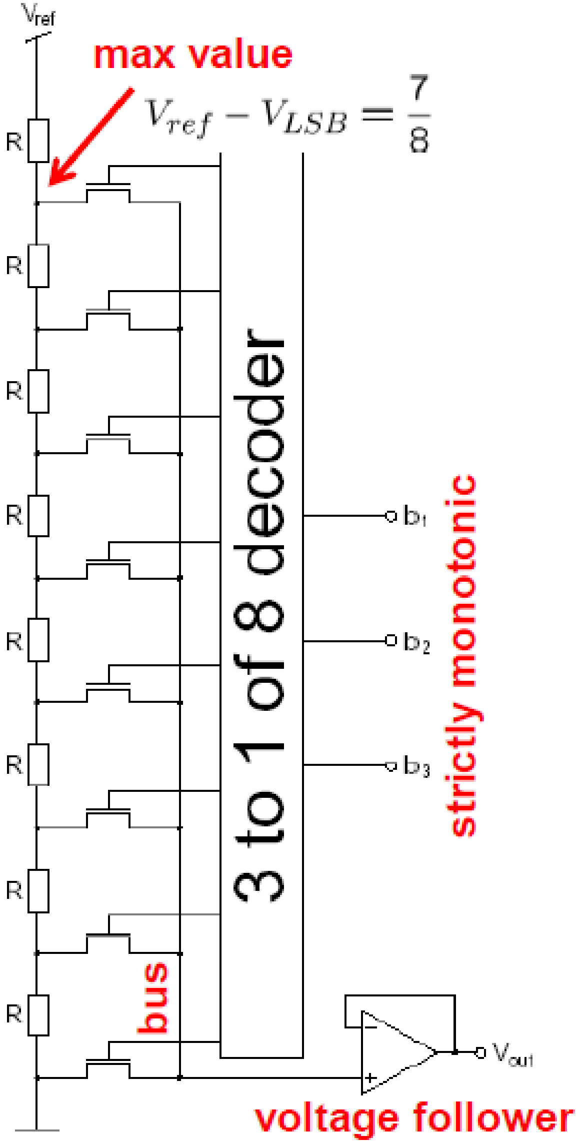
N κόμβοι μεταφοράς τάσης (N switch delay paths)

Μέχρι 10 bits

Ελαχιστοποίηση των glitches

2expN resistors

Resistor-string DAC with digital decoding



High-speed implementation (when compared to the previous one), due to maximum of one switch in series

• Less resistance through switches

• The switches are controlled by digital logic

• More area for the decoder compared to the previous DAC

• Larger capacitance on the buffer input, due to the 2 expN transistors connected to it

• Pipelining may be applied for "moderate speed"

• 2 exp N resistors are required

Resistor-String D/A Converters

- One of the first integrated MOS 8-bit DAC.
- Switch = pass transistor or CMOS transmission gate.
- Accuracy $\Leftarrow$ matching precision of R
  Polysilicon resistor $\Rightarrow 20 \sim 30\ \Omega/\Box$, 10-bit accuracy.
- Guaranteed monotonicity for voltage-insensitive V_{OS} of the buffer.
- Delay through the switch network: $\tau \approx RCn^2/2 = \sum R_{oi}C_i$.
- Tree decoding or digital decoding (2^N junctions on the output line).

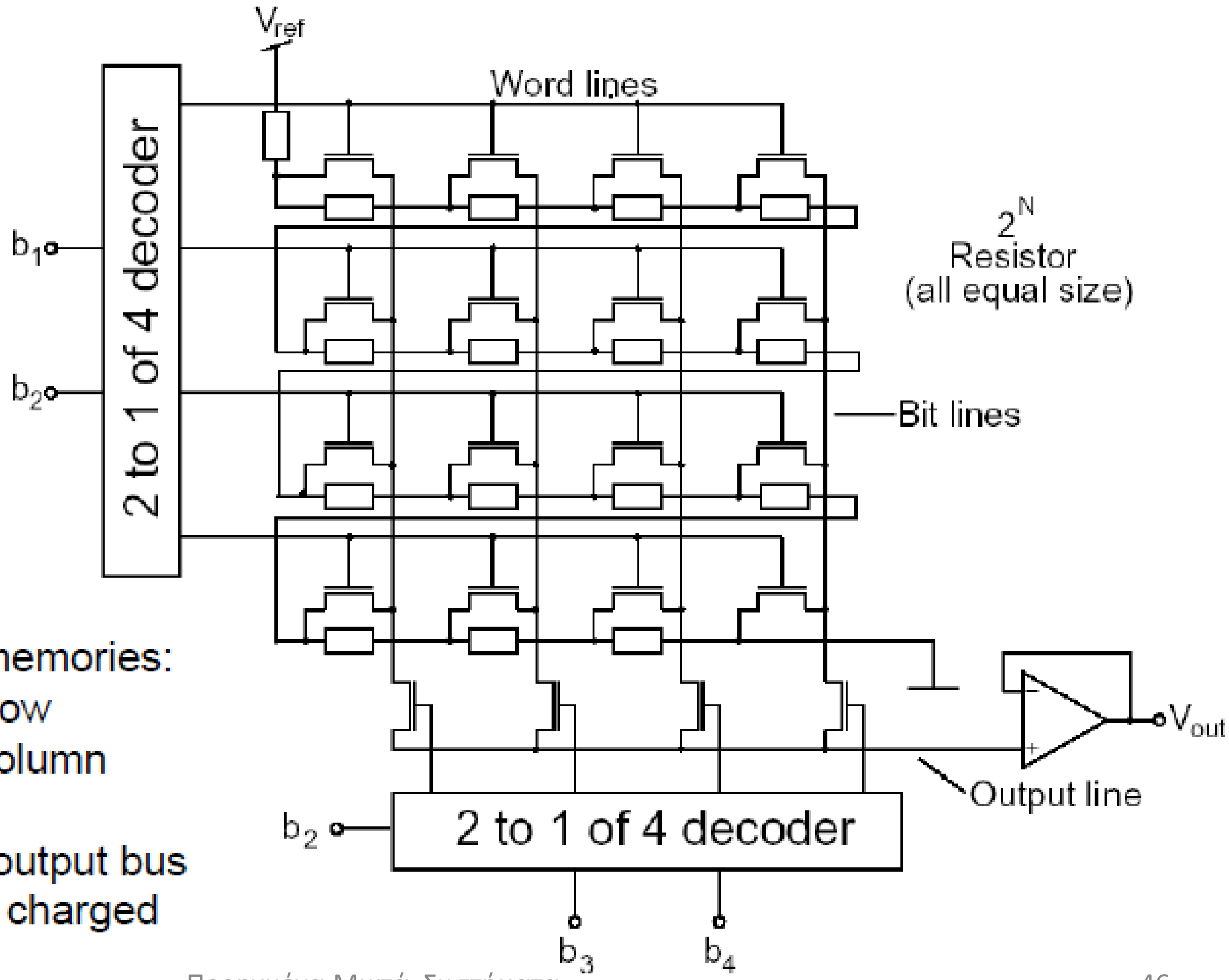
2expN resistors are required (when only one resistor string is included)

Delay through switch network is the major speed limitation of the circuit

Minimization of glitches – 10 bits at most

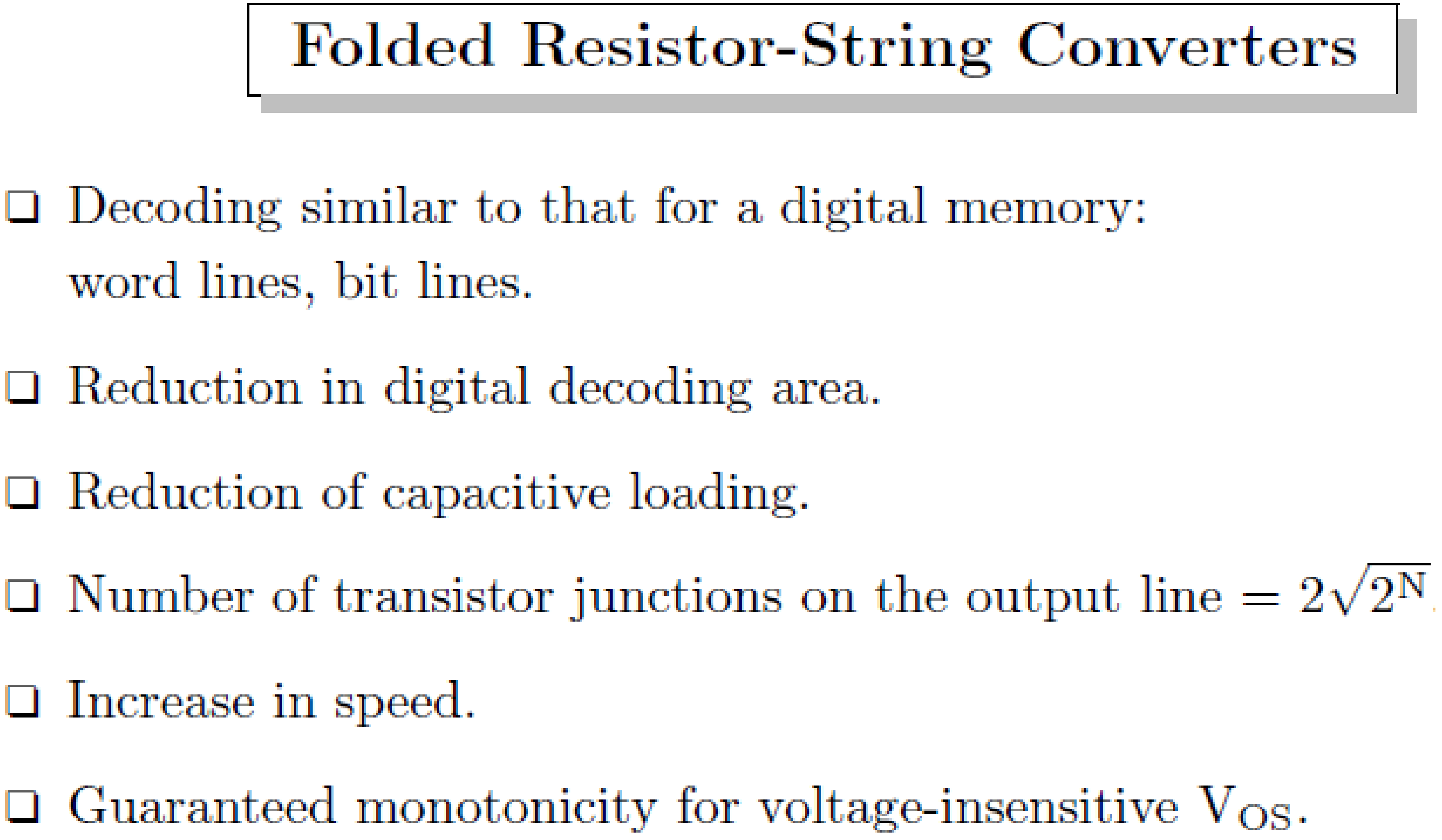
Folded Resistor String Converter

Combine the advantages of both converters:
(low effort for decoder, small load cap.)



Access scheme as in memories:
MSBs select row
LSBs select column

$2\sqrt{2^N}$ transistors at output bus
all bitlines are charged



Folded Resistor-String Converters

□ Decoding similar to that for a digital memory: word lines, bit lines.

□ Reduction in digital decoding area.

□ Reduction of capacitive loading.

□ Number of transistor junctions on the output line = $2\sqrt{2^{N}}$.

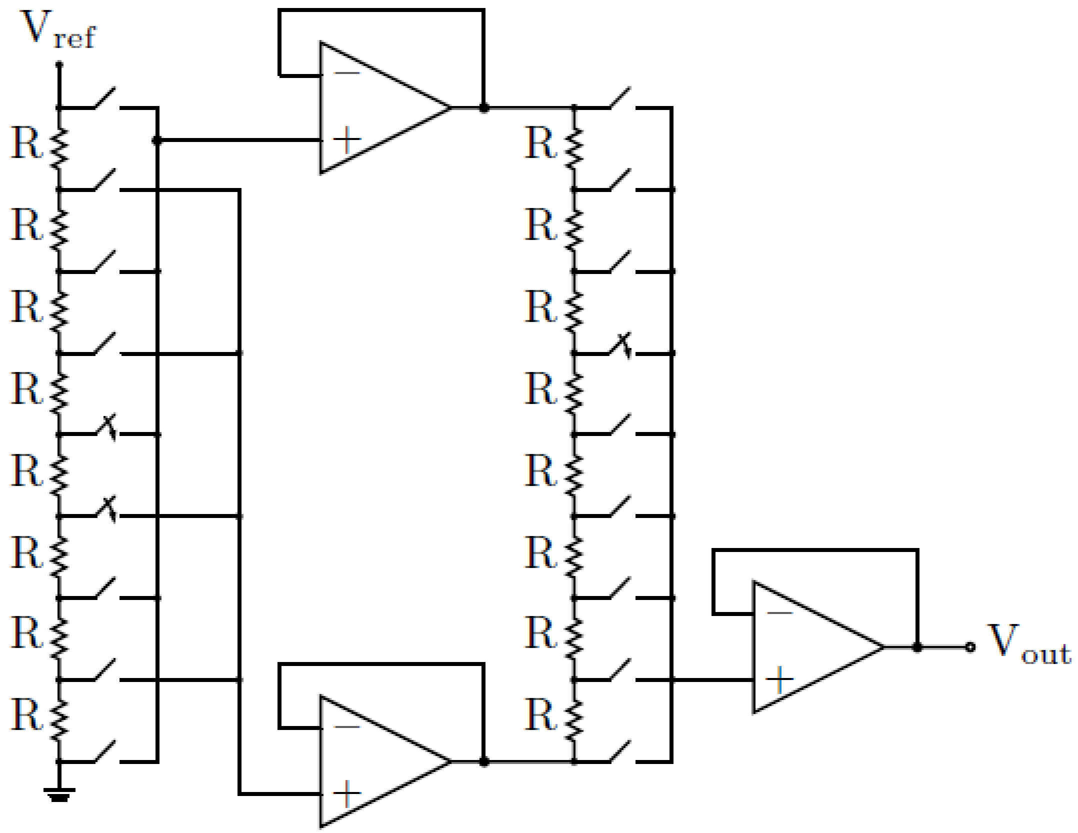
□ Increase in speed.

□ Guaranteed monotonicity for voltage-insensitive V_{OS}.

Example of Number of transistor junctions reduction at the output line:

- 4 bit case: 8 instead of 16
- 8 bit case: 32 instead of 256
- 10 bit case: 64 instead of 1024

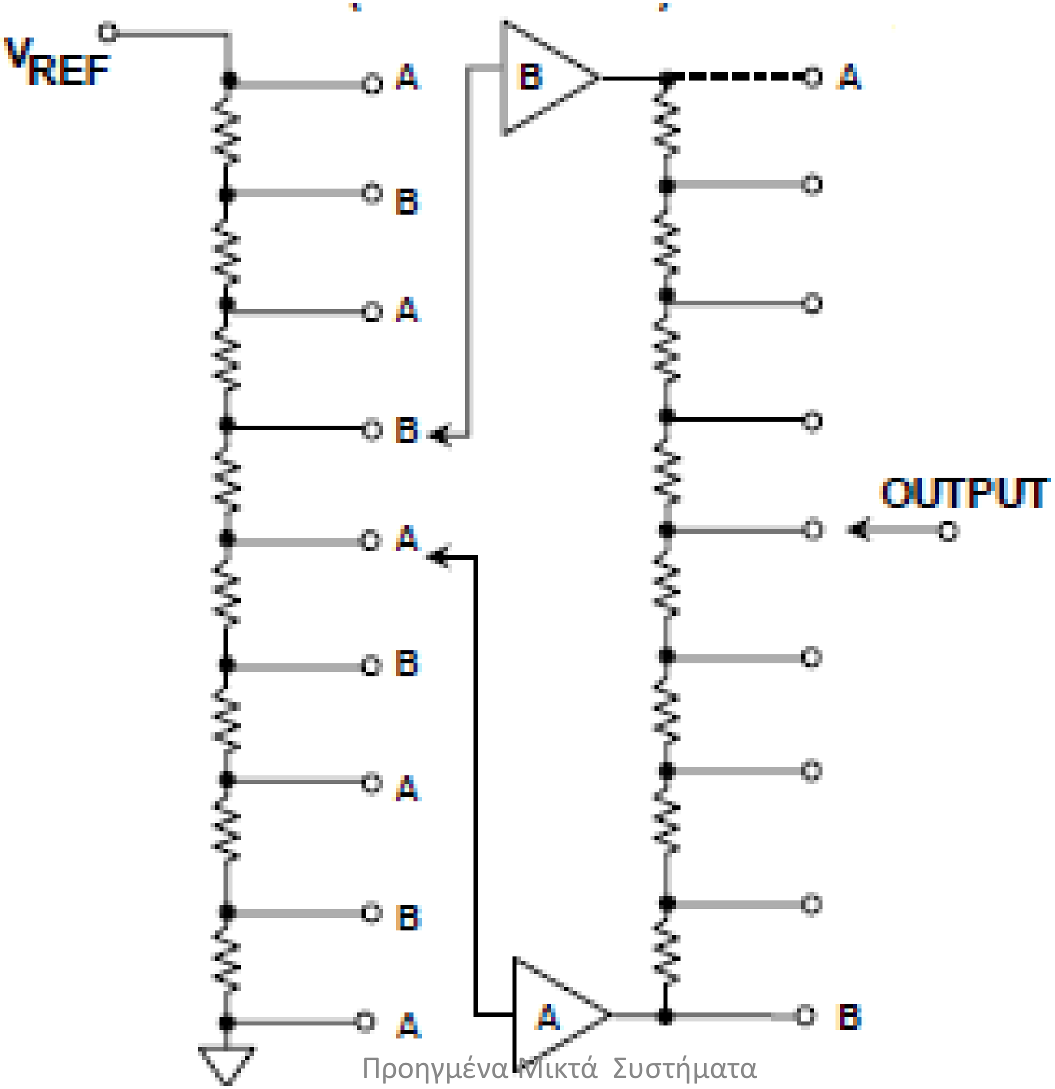
Multiple R-String



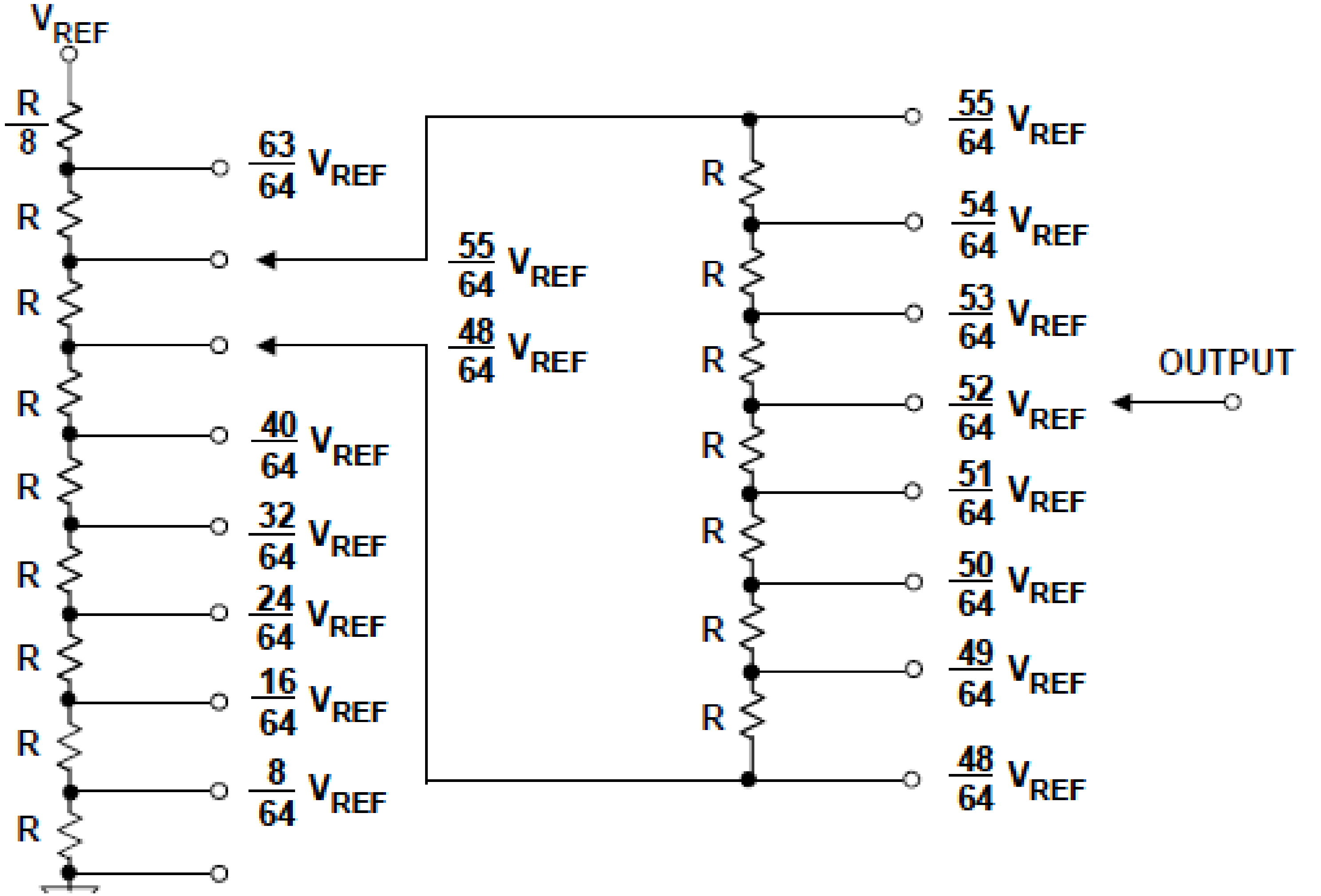
Basic Concept

- Subdivide voltage range in coarse sub-intervals first
- Copy the respective voltage interval
- Fine interpolation of the copied interval

Multiple R-String
(Basic Mechanism demo)



Multiple R-String
(Basic Mechanism demo)

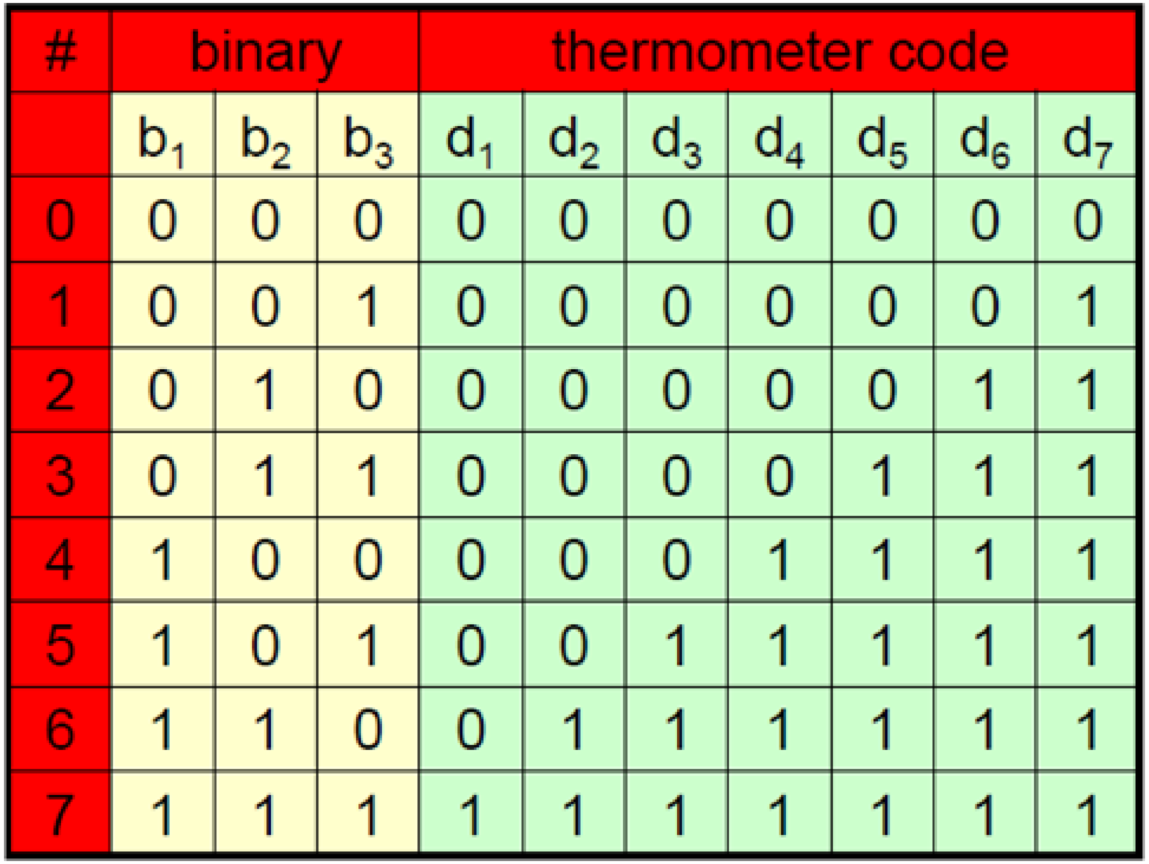


Multiple R-String

- A second tapped resistor string is connected between buffers whose inputs are two adjacent nodes of the first resistor string, as shown.
- In the 6-bit case the 3 MSBs determine the two adjacent nodes. The 2nd (“fine”) string linearly interpolates between the two adjacent voltages from the first (“coarse”) resistor string
- Additional logic needed to handle polarity switching, related to which intermediate buffer has the highest voltage on the input
- Guaranteed monotonicity assuming matched opamps and voltage insensitive offset voltages
- 2 x 2N/2 resistors are required
- Relaxed matching requirements for the 2^{nd} resistor string.

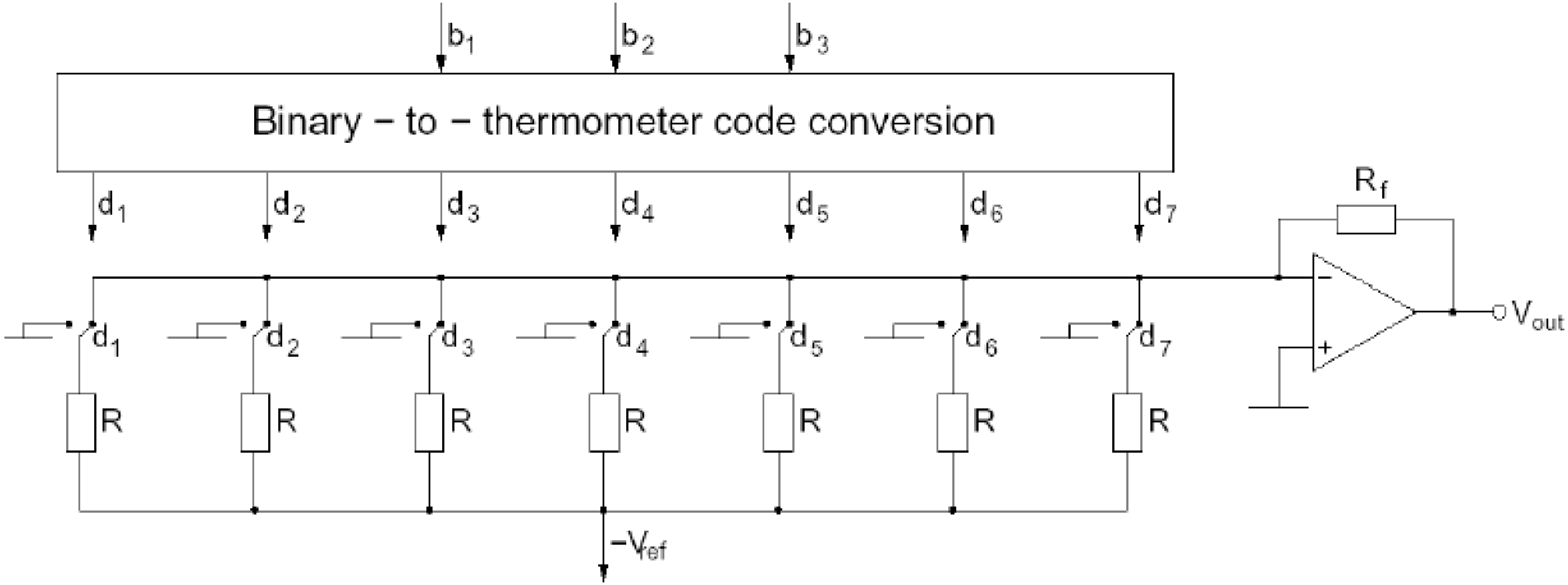
Ex.: 10 bit, 4 bits for the 1st string, matched to 0.1 %. Requirements for 2nd string much more relaxed , e.g.= 1.6 %

Thermometer Code Converters (method to force monotonicity)



#	binary			thermometer code						
	b_1	b_2	b_3	d_1	d_2	d_3	d_4	d_5	d_6	d_7
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	1	1
3	0	1	1	0	0	0	0	1	1	1
4	1	0	0	0	0	0	1	1	1	1
5	1	0	1	0	0	1	1	1	1	1
6	1	1	0	0	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1

Thermometer Code Converters (method to force monotonicity)

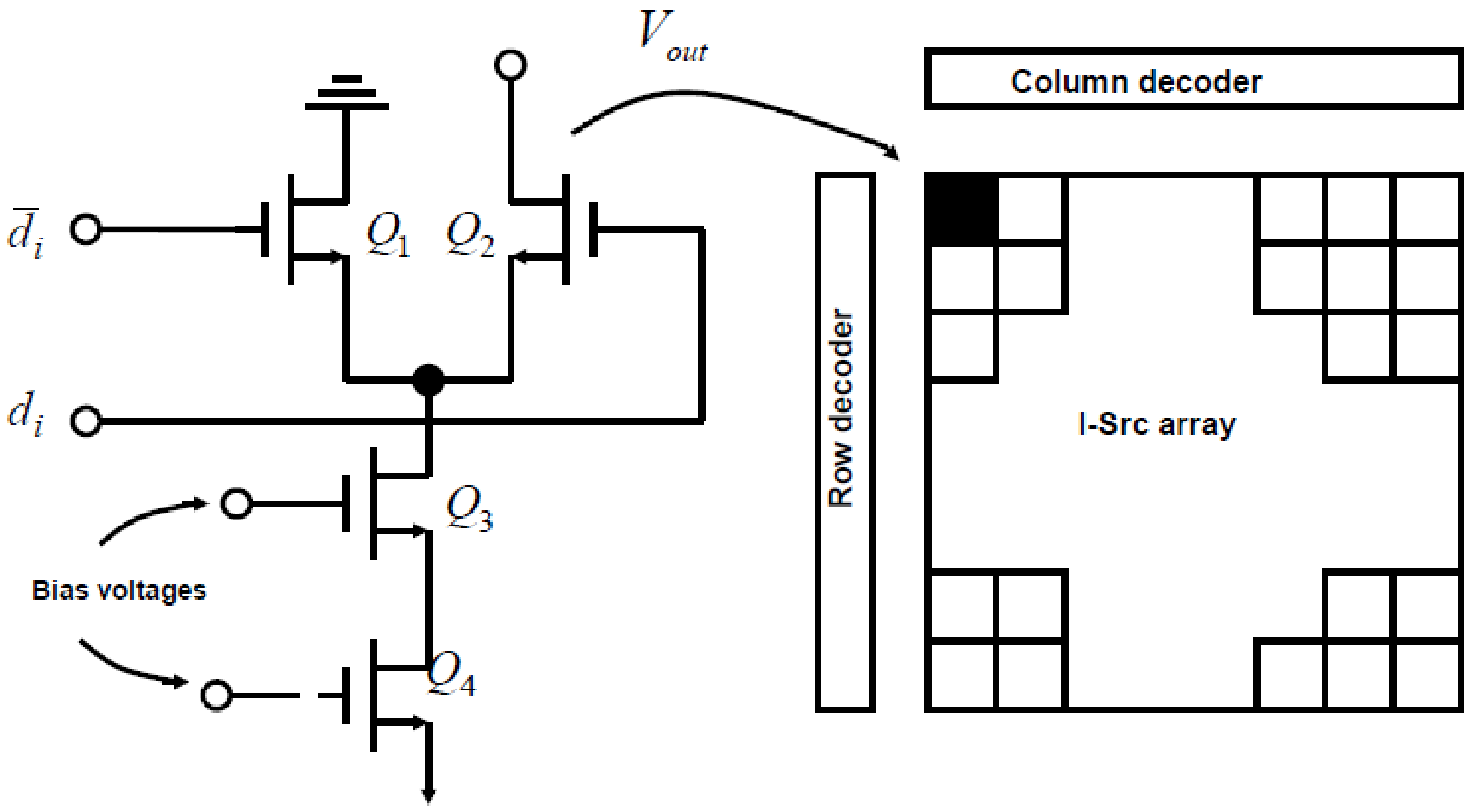


Main drawback of thermometer –coded architecture:
For every LSB a it is needed : a current source ,a switch, and a complex decoding circuit as well as a binary to thermometer decoder =>impractical for more than 10 bits

Thermometer-Code Converters

- Thermometer code: difference in one bit from next codes.
- Not a minimal representation: $2^N - 1$ bits for 2^N digital values.
- Low DNL errors: output change by only 1 LSB.
- Guaranteed monotonicity: never-lower output change.
- Reduced glitching noise:
  when the input code changes from $011 \cdots 11$ to $100 \cdots 00$.
- Same area of the analog circuitry as binary-weighted approach: since the area of resistors is proportional to their size.
- All equal-size switches since they pass equal currents.

Thermometer-Code Current-Mode DAC (Cont.)



Thermometer-Code Current-Mode Converters

- Basis for a variety of designs.
- Matrix current sources with equal value.
- Thermometer-code decoders for row and column decoding.
- Off-chip 50 Ω or 75 Ω load for high speed.
- Inherent monotonicity.
- Low DNL errors.
- Need for precise timing

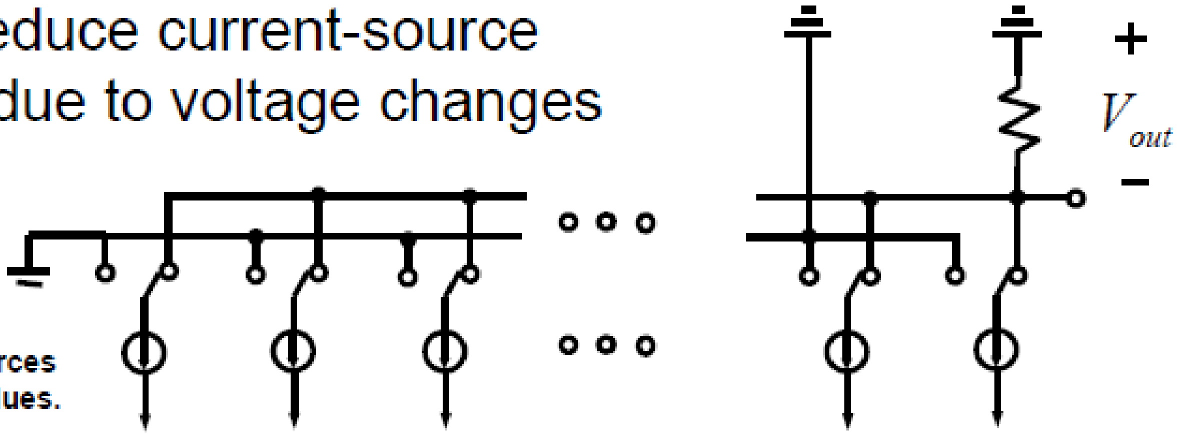
Thermometer-Code Current-Mode DAC

- Row and column decoders
- Inherent monotonicity
- Good DNL errors

  INL errors depend on the placement of the current sources
- In high-speed applications

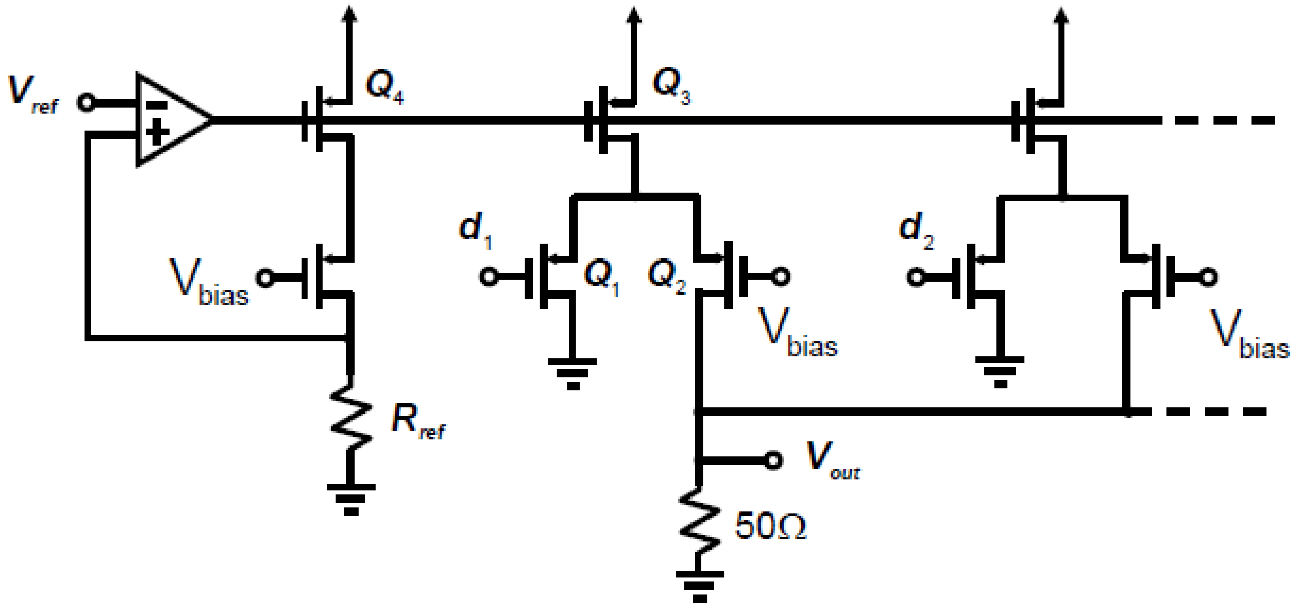
  1. The output current feeds directly into an off-chip 50 Ω or 75 Ω resistor, rather than an output OPAMP.

2. Cascode current sources are used to reduce current-source variation due to voltage changes in V_{out}



All current sources are of equal values.

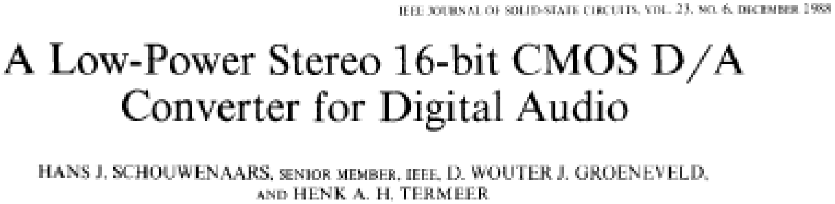
Thermometer-Code Current-Mode DAC (Cont.)



- Can be clocked at the maximum rate without the need for precisely timed edges
- Q_2 and Q_3 effectively form a cascode current source when they drive current to the output.
- To maximize speed, the voltage swing at the common connection (e.g. Q_1, Q_2 and Q_3) of the current switches should be small.

Dynamically Matched Current Sources (12.3)

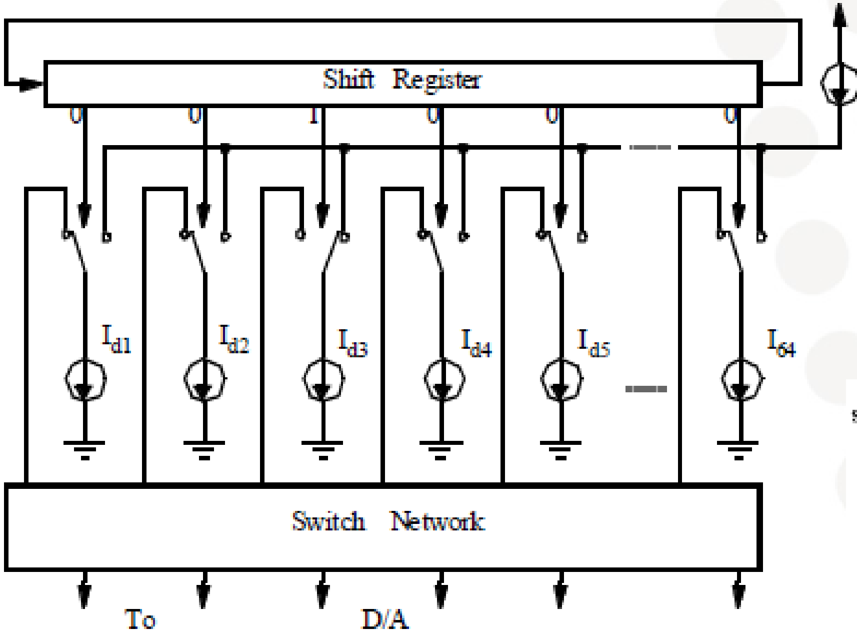
- for high resolution



IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 23, NO. 6, DECEMBER 1988

A Low-Power Stereo 16-bit CMOS D/A Converter for Digital Audio

HANS J. SCHOUWENAARS, SENIOR MEMBER, IEEE, D. WOUTER J. GROENEVELD, AND HENK A. H. TERMEER



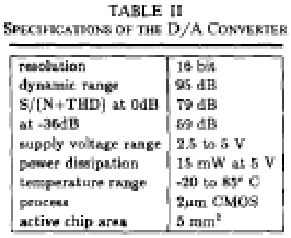


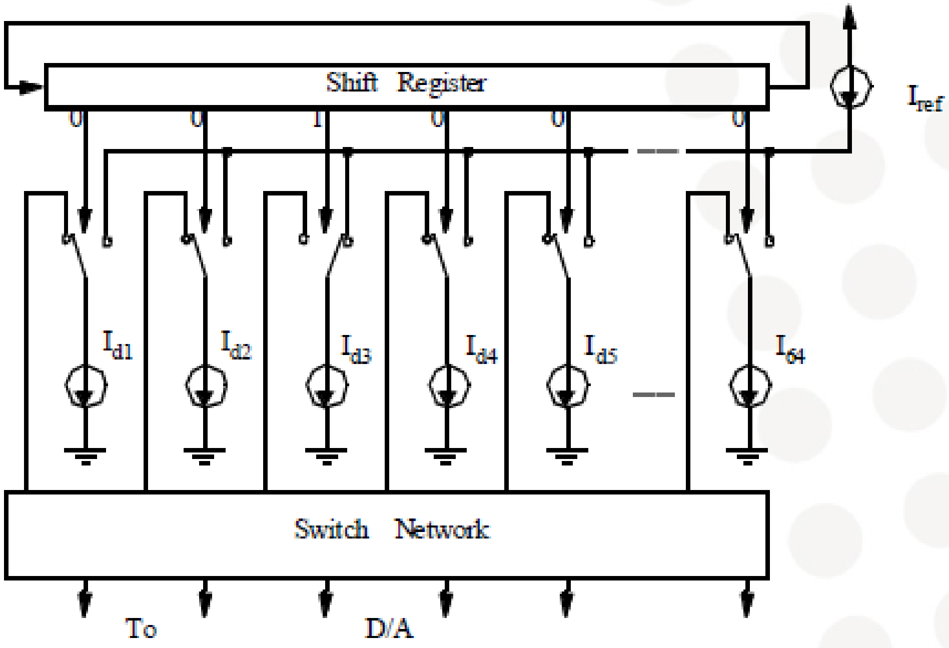
TABLE II
SPECIFICATIONS OF THE D/A CONVERTER

resolution	16 bit
dynamic range	95 dB
S/(N+THD) at 0dB	79 dB
at -36dB	59 dB
supply voltage range	2.5 to 5 V
power dissipation	15 mW at 5 V
temperature range	-20 to 85° C
process	2μm CMOS
active chip area	5 mm²

- Current sources are periodically being regulated to ideally the same value (matched) during normal operation, to ensure proper resolution.
- A “once and for all” matching of each current source is not enough due to mechanisms including temperature drift and gate leakage.

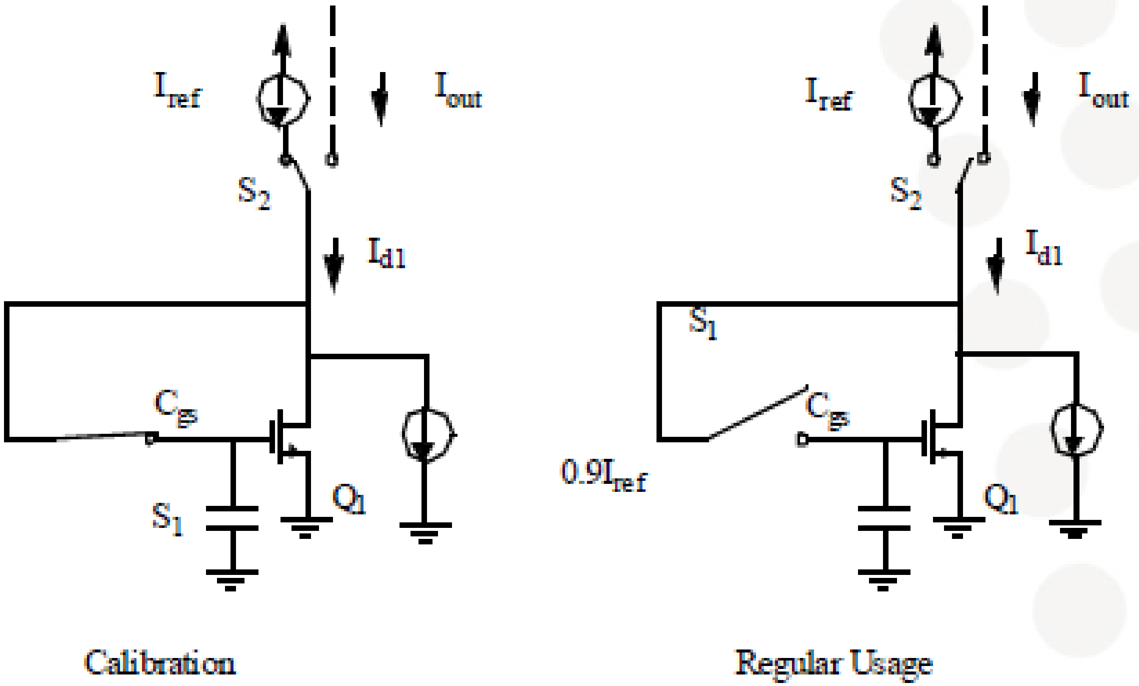
Dynamically Matched Current Sources

(12.3)



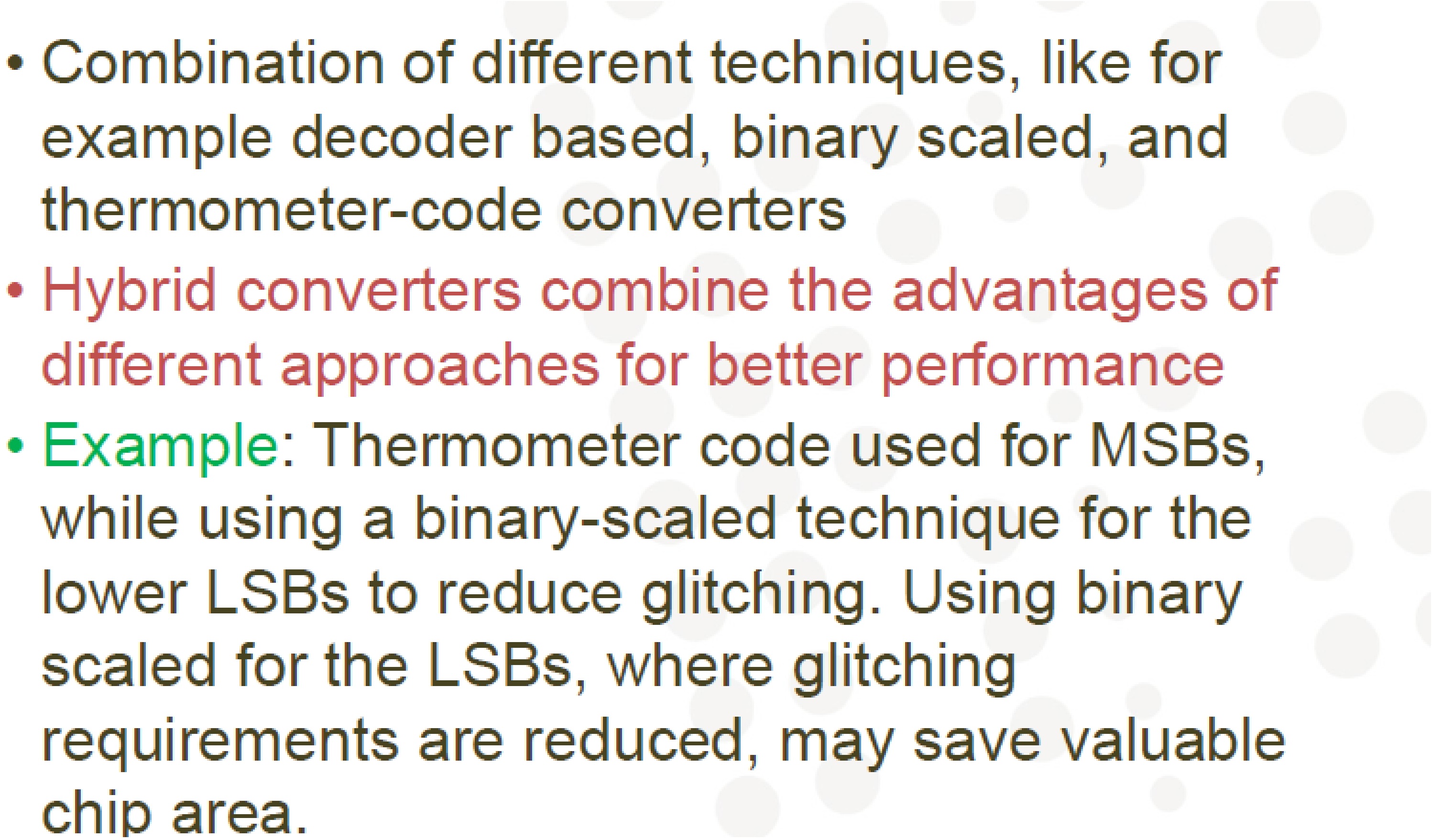
- 6 MSB realized using a thermometer code. (binary array for the remaining bits)
- All currents are matched against $I_{ref,}$ one after one, to get the same precise value on all Idi.
- One extra current source is included to provide continous operation, even when one of the sources is being calibrated.

Dynamic matched current sources– method for calibration



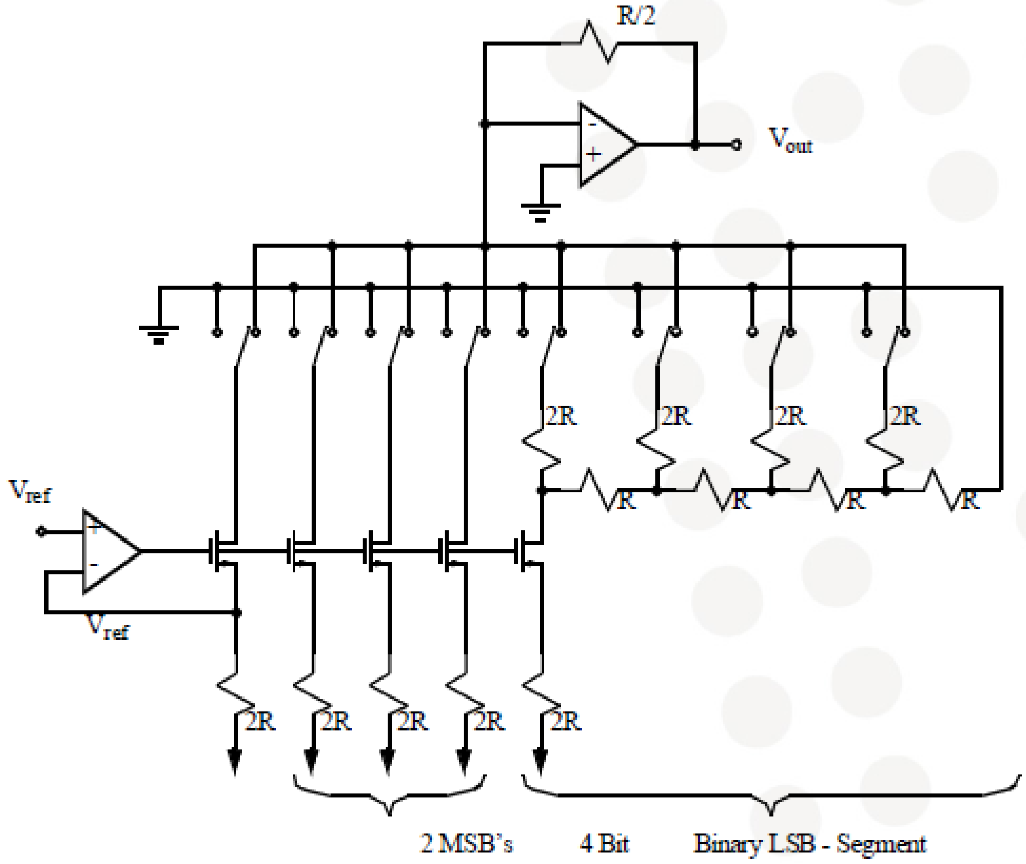
- I $_{di}$ is connected to I_{ref} during calibration. This places whatever voltage is necessary across the parasitic C_{gs} so that I_{di} equals I_{ref} . When S_i is opened Idi remains nearly equal to I_{ref}.
- Using $0.9I_{ref}$ in parallell makes Q_1 need only to source a current near 0.1 I_{ref}. ' ıt

Hybrid Converters



- Combination of different techniques, like for example decoder based, binary scaled, and thermometer-code converters
- Hybrid converters combine the advantages of different approaches for better performance
- Example: Thermometer code used for MSBs, while using a binary-scaled technique for the lower LSBs to reduce glitching. Using binary scaled for the LSBs, where glitching requirements are reduced, may save valuable chip area.

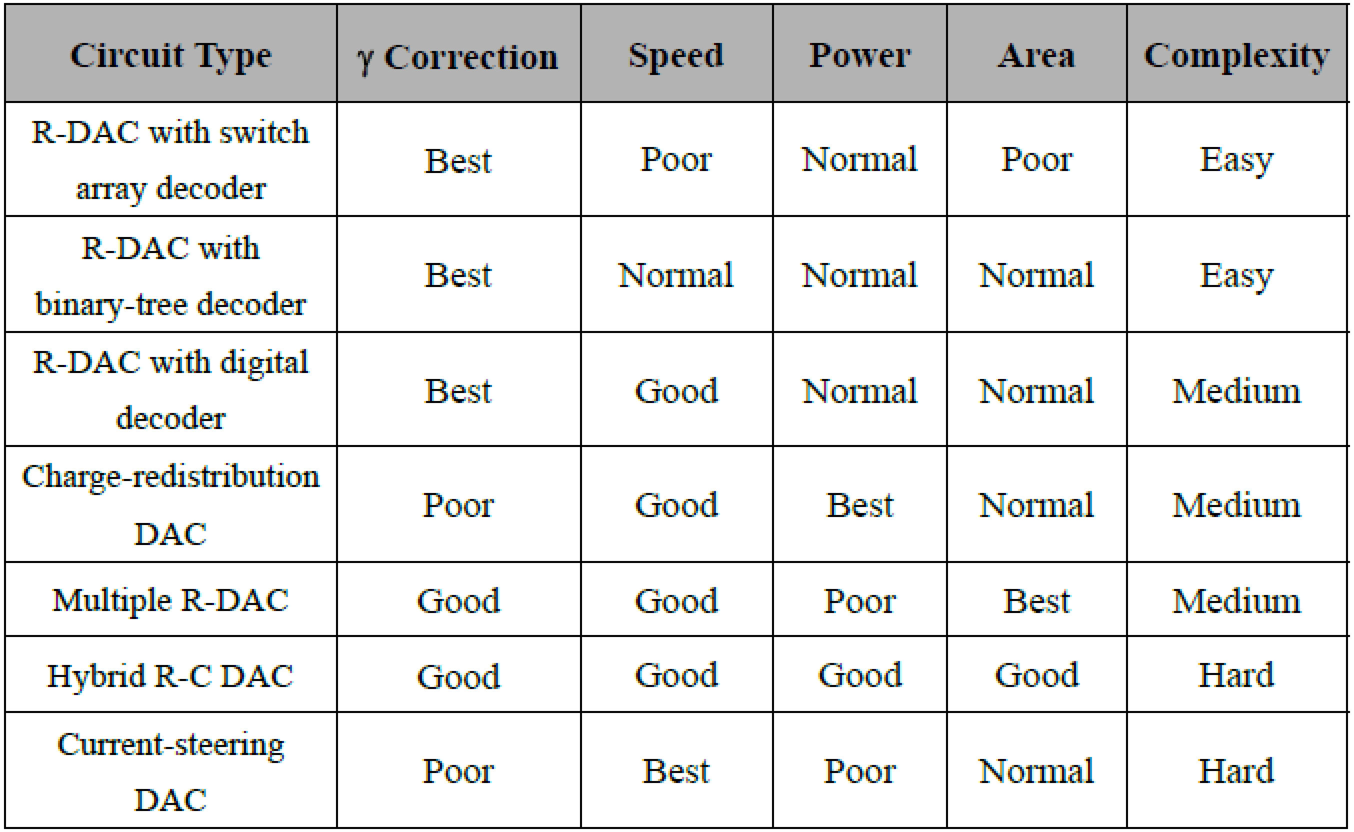
Hybrid (Segmented) Converters



- Combination of thermometer- and binary
- 2 MSB's are thermometer (reduces glitches)
- 4 LSB's are binary

Comparison of various D/A Techniques

Comparisons of many kinds of the digital-to-analog converter circuits



Circuit Type	γ Correction	Speed	Power	Area	Complexity
R-DAC with switch array decoder	Best	Poor	Normal	Poor	Easy
R-DAC with binary-tree decoder	Best	Normal	Normal	Normal	Easy
R-DAC with digital decoder	Best	Good	Normal	Normal	Medium
Charge-redistribution DAC	Poor	Good	Best	Normal	Medium
Multiple R-DAC	Good	Good	Poor	Best	Medium
Hybrid R-C DAC	Good	Good	Good	Good	Hard
Current-steering DAC	Poor	Best	Poor	Normal	Hard

A/D CONVERTERS

Applications: ADC Performance is Critical

- Dramatic improvement in converter performance is required for emerging IEEE communication standards
- Data converters will be a key enabling technology to realize ICs at the appropriate power

Broadband Wireless

Disc Drives

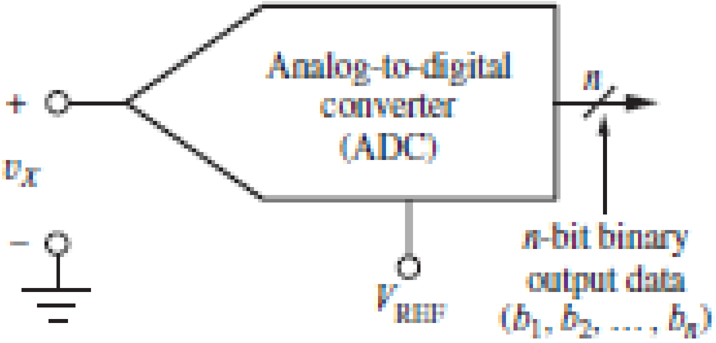
Mobile Phones

ADC

Base Stations

Networking

A/D Converters



Block diagram representation for an A/D converter

The **analog-to-digital converter, also known as an A/D converter** or **ADC, is used to transform analog information in electrical form into digital data.**

The ADC in above Fig. takes an unknown continuous analog input signal, most often a voltage υχ *, and converts it* into an *n-bit binary number*

The n-bit number is a binary fraction representing the ratio between the unknown input voltage *vx and the converter's* full-scale voltage $V_{FS} = K\, V_{REF}$.

– Output = 2expn x GAIN / VREF

- n = # of Output Bits (Resolution)
- K = Gain Factor (usually "1")
- GAIN = Analog Input Voltage (or Current)
- VREF (IREF)= Reference Voltage (or Current)

Characteristics of ADC and DAC

- ADC

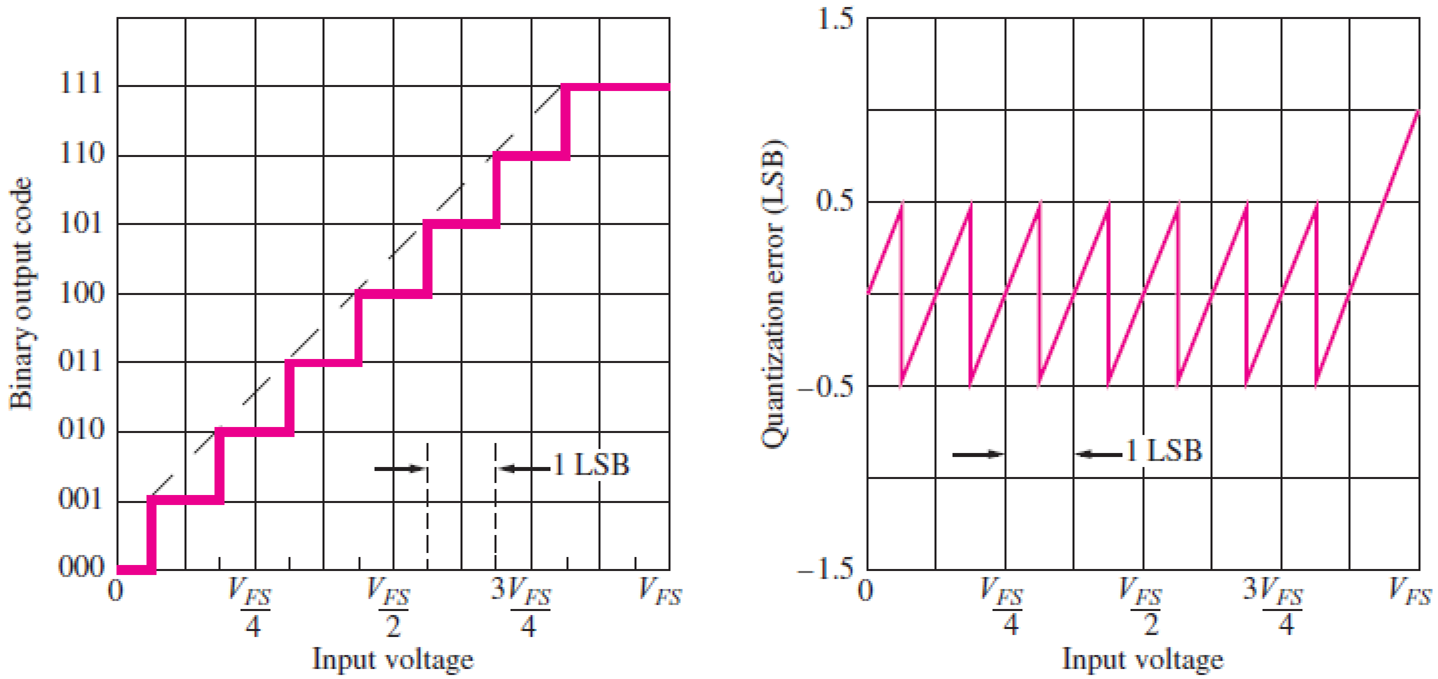
Similar Characteristics to DACs

- Offset , gain error , DNL and INL
- Quantization Noise
- Sampling-time uncertainty +

And additionally:

- missing code +
- a number of features like SNR , SINOD, ENOB, FOM etc.

ADC Characteristics



ADC 3-bit graphs showing (ideal) ADC transfer curve and quantization noise

Quantization error , υε , lies somewhere within 1 LSB quantization interval and followσ uniform distribution

$$v_\varepsilon = |v_X - (b_1 2^{-1} + b_2 2^{-2} + \cdots + b_n 2^{-n}) V_{FS}|$$

ADC Characteristics- DNL and INL errors

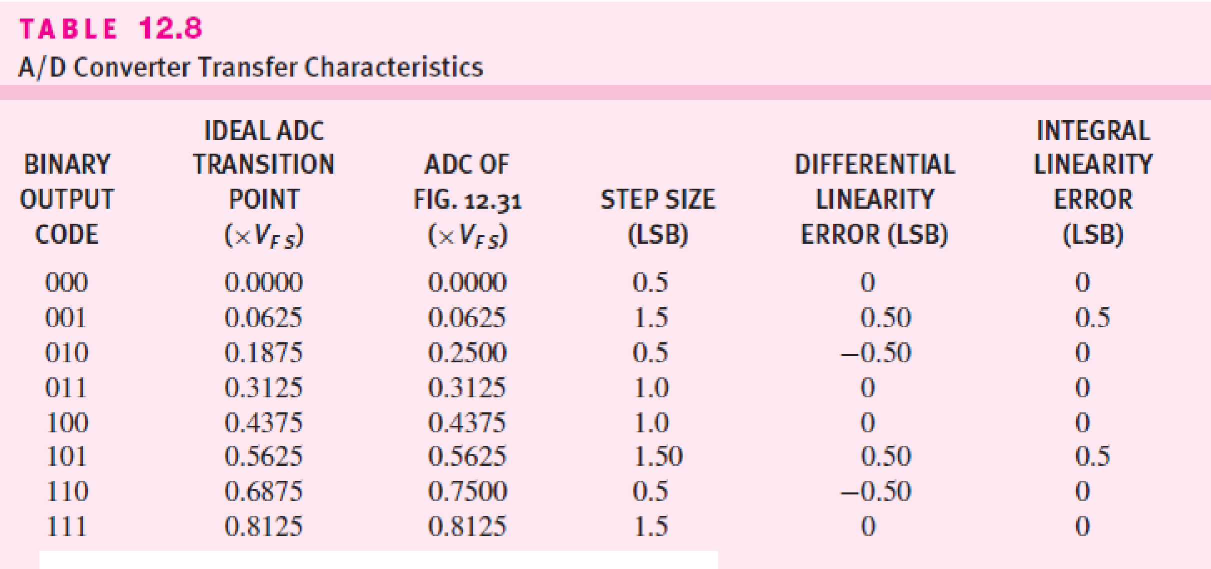
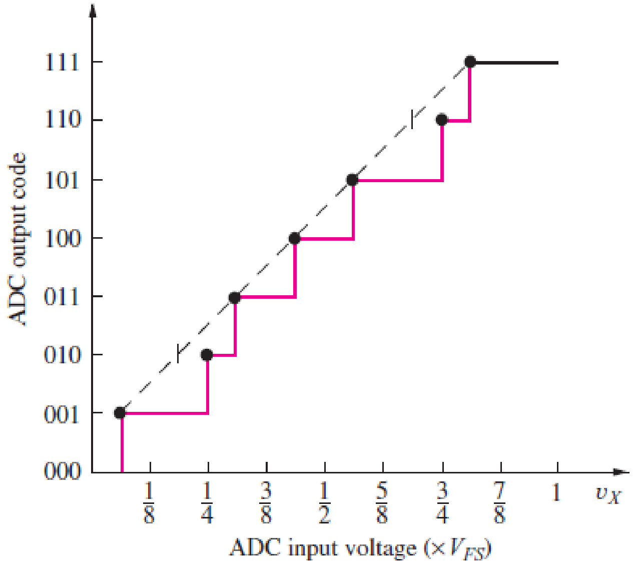


TABLE 12.8

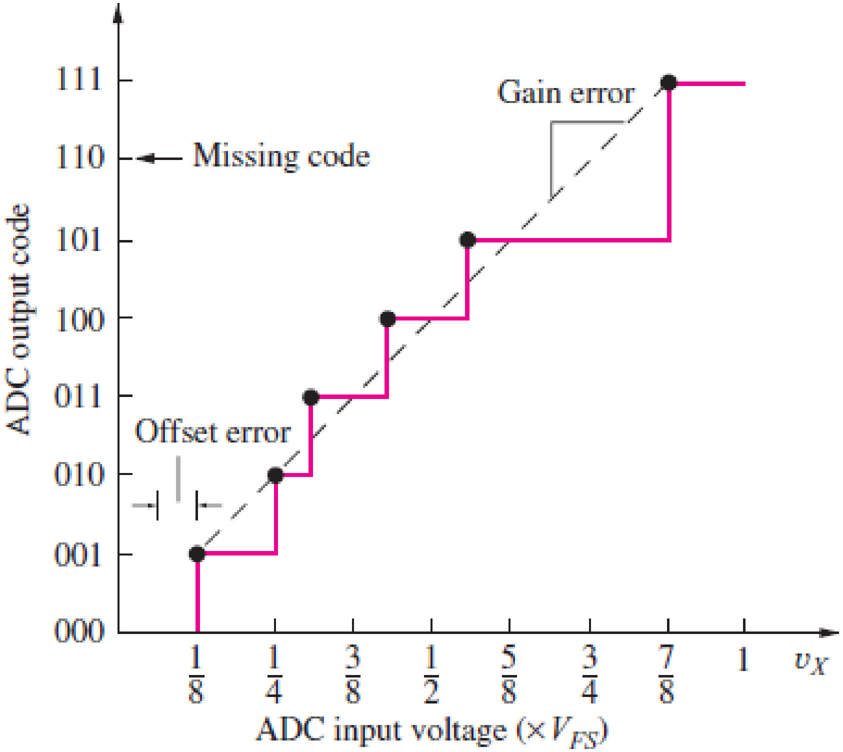
A/D Converter Transfer Characteristics

BINARY OUTPUT CODE	IDEAL ADC TRANSITION POINT ($\times V_{FS}$)	ADC OF FIG. 12.31 ($\times V_{FS}$)	STEP SIZE (LSB)	DIFFERENTIAL LINEARITY ERROR (LSB)	INTEGRAL LINEARITY ERROR (LSB)
000	0.0000	0.0000	0.5	0	0
001	0.0625	0.0625	1.5	0.50	0.5
010	0.1875	0.2500	0.5	−0.50	0
011	0.3125	0.3125	1.0	0	0
100	0.4375	0.4375	1.0	0	0
101	0.5625	0.5625	1.50	0.50	0.5
110	0.6875	0.7500	0.5	−0.50	0
111	0.8125	0.8125	1.5	0	0



An example of A non-ideal 3-bit ADC with DNL and INL Errors-shown also in above Table.Definitions of both these errors=identical to ones given in DACs

ADC Characteristics-
gain , offset errors and missing codes



The slope of the fitted line does not give 1 LSB = *VFS/8, so the converter also exhibits a gain* error.

A new type of error, which is specific to ADCs, can be observed in Figure The output code jumps directly from 101 to 111 as the input passes through 0*.875VFS.*

The output code 110 never occurs, so this converter is said to have a **missing code**

A converter with a differential linearity error of less than 1 LSB does not exhibit missing codes in its input-output function.

An ADC can also be **nonmonotonic. If the output code decreases as the input voltage increases, the converter has a** nonmonotonic input–output relationship.

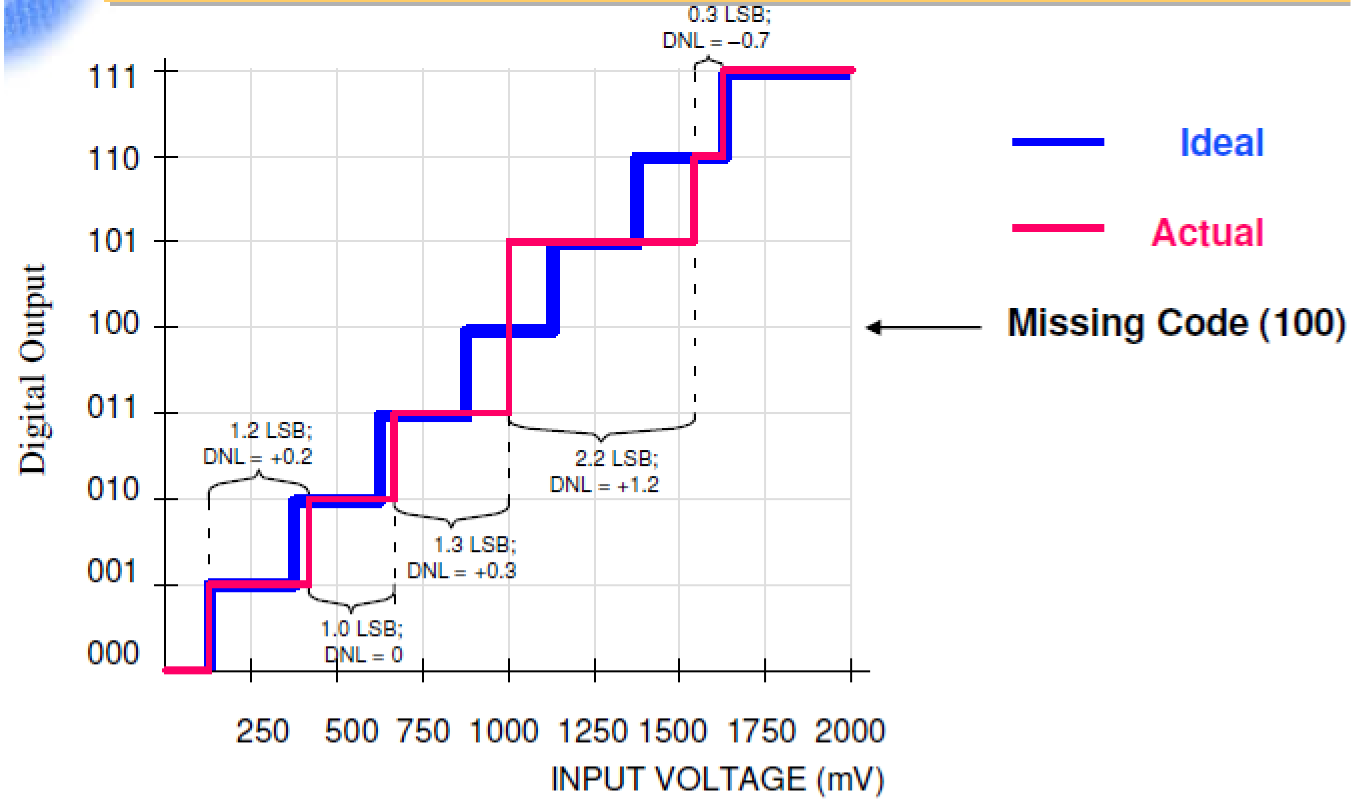
All these deviations from ideal A/D (or D/A) converter behavior are temperature-dependent; hence, converter specifications include temperature coefficients for gain, offset, and linearity.

A good converter will be monotonic with less than 0.5 LSB differential linearity error and no missing codes over its full temperature range.

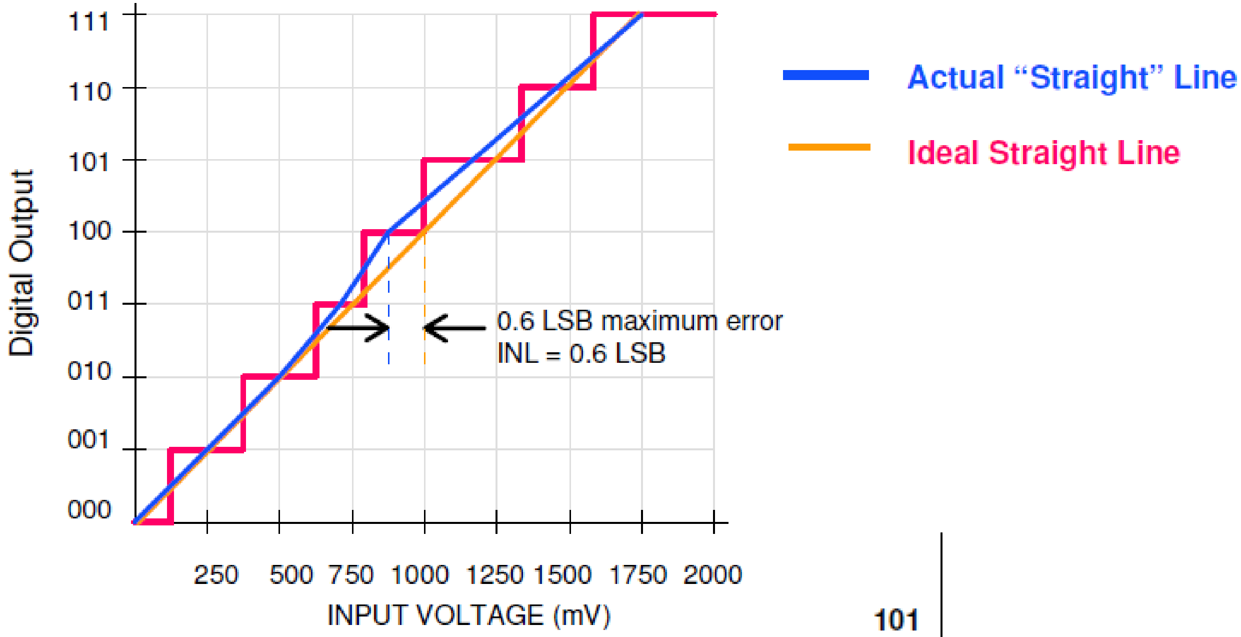
ADC Characteristics-

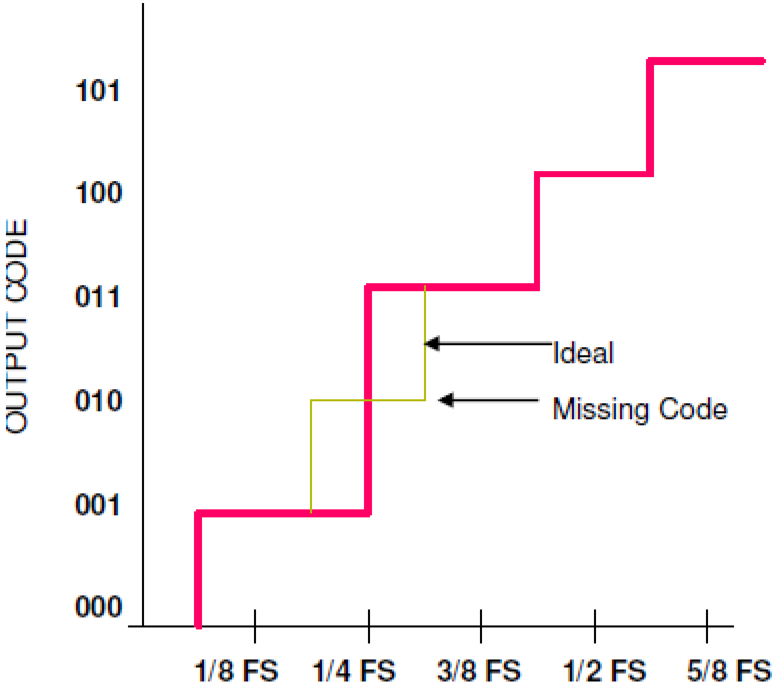
some examples with various ADC errors described previously

DNL

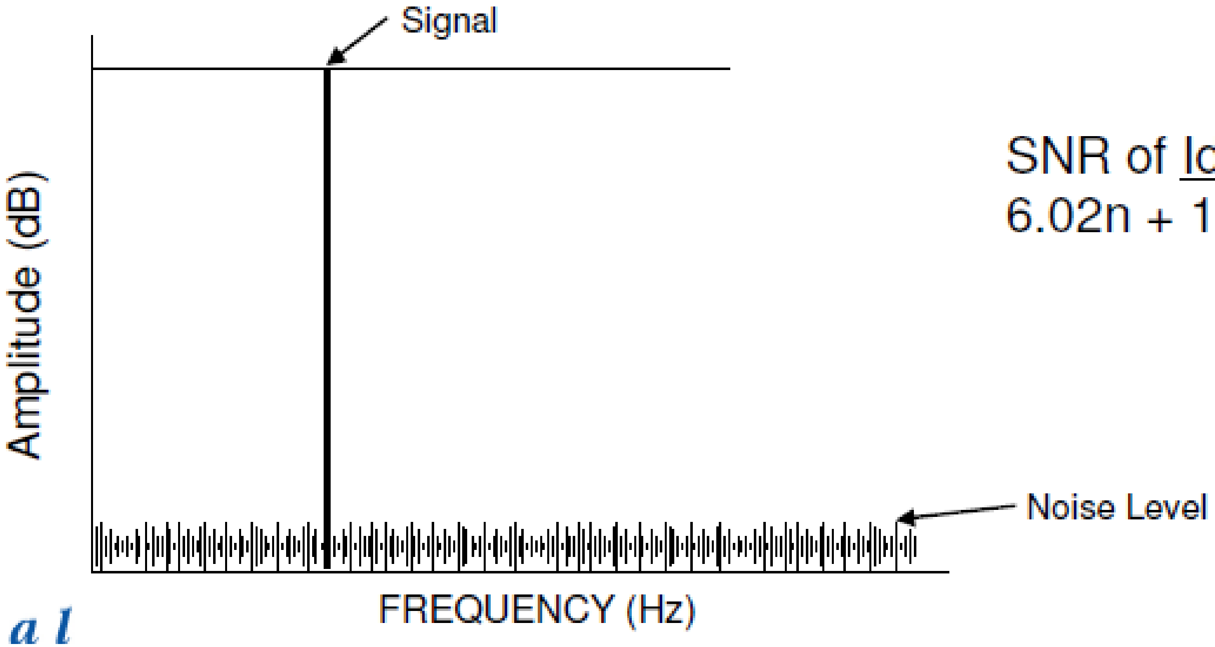


ADC Characteristics-examples with various errors





SNR(Signal to Noise Ratio)



SNR of Ideal ADC = 6.02n + 1.76 dB

Signal-to-Noise Ratio (SNR) is the ratio of the output signal amplitude to the output noise level, not including harmonics or dc. A signal level of 1VRMS and a noise level of 100µVRMS yields an SNR of 80dB.

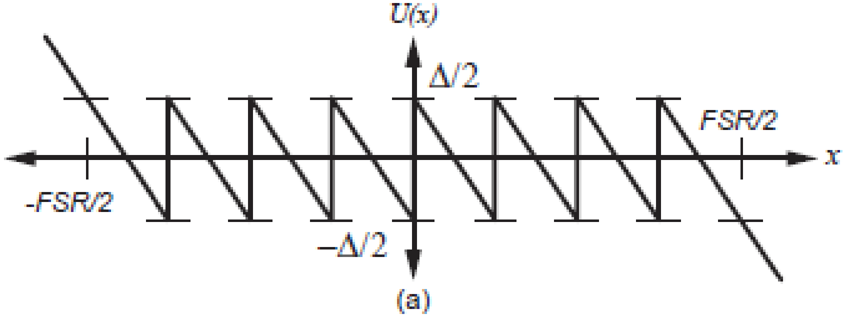
SNR usually degrades as frequency increases because the accuracy of the comparator(s) within the ADC degrades at higher input slew rates. This loss of accuracy shows up as noise at the ADC output.

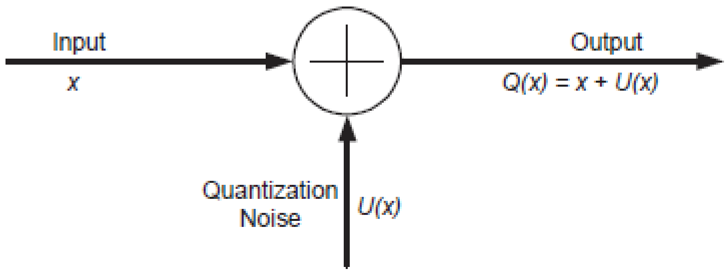
In an A/D converter, noise comes from 3 sources: (1) quantization noise, (2) noise generated by the converter itself and (3) jitter

SNR increases with increasing input amplitude until the input gets close to full scale

SNR(Signal to Noise Ratio)

- Calculation of quantization noise and SNR of an ideal ADC (with quantization noise only)





- *U(x) = Q(x) − x* Since the quantization noise *U(x)* is assumed to be uniformly distributed on (− Δ /2, Δ/ 2) the output noise power can be easily calculated as:

$$\sigma^2(\Delta)=\int_{-\infty}^{+\infty}x^2 p_\Delta(x)dx \quad =\int_{-\Delta/2}^{+\Delta/2}x^2\frac{1}{\Delta}dx \quad =\frac{\Delta^2}{12}$$

The power of the full swing sinusoidal input signal is: $P_s=(FSR/2)^2/2=FSR^2/8=(2^N\Delta)^2/8$

The quantizer SNR is therefore given by $SNR_Q=10\log\left(\frac{2^{2N}\cdot\Delta^2/8}{\Delta^2/12}\right)$ =10 log(3/2 x 2^{2N})=

- $=6.02N+1.76(dB)$

This is a frequently used equation for predicting optimum A/D performance. For a 7-bit converter maximum SNR is 43.9 dB, and for an 8-bit converter the maximum SNR is 49.92dB

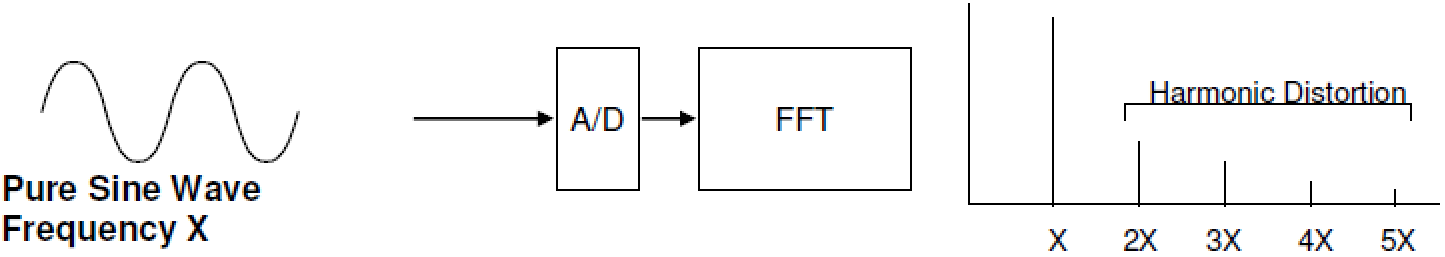
ADC Characteristics-ENOB

- Equation for SNR_Q=6.02N +1.76 can be used to assess the performance of any ADC relative to the ideal. By replacing the maximum achievable SNR by the actual SNR and solving for the equivalent resolution, *N, a figure of merit called the Effective-Number-Of-Bits (ENOB) r*esults in:

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

- ENOB says that the converter performs as if it were a theoretically perfect converter with a resolution of ENOB bits
- The effective-number-of-bits is a commonly used metric for summarizing the performance of non-ideal quantizers.
- In practice, A/D converters encounter inputs which are more complicated than simple sinusoids
- ENOB degrades as frequency increases and as input level decreases for the same reasons that SNR degrades with frequency increase and improves as input level increases.

ADC Characteristics-
THD – Total Harmonic Distortion



THD is the ratio of the rms total of harmonic components to the RMS value of the output signal and relates the RMS sum of the amplitudes of the harmonics to the amplitude of the fundamental

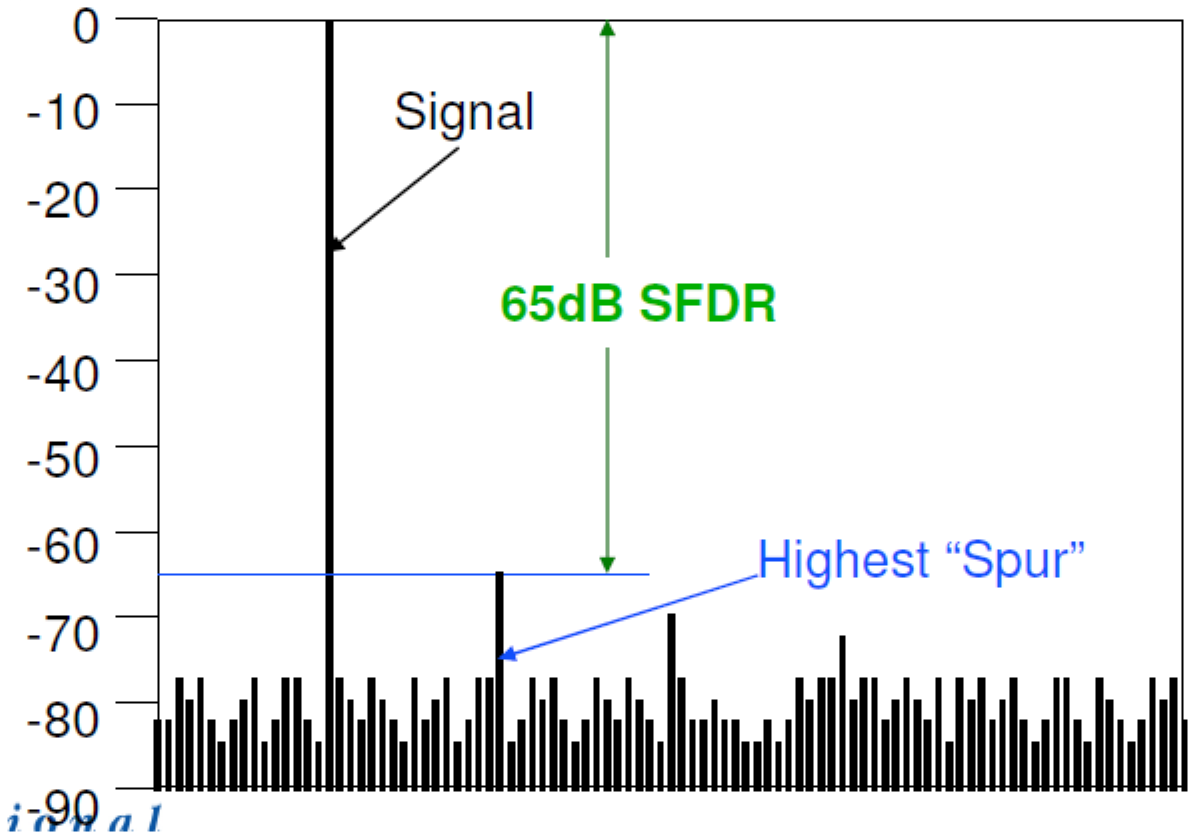
THD gives an indication of a circuit’s linearity in terms of its effect on the harmonic content of a signal . It is defined as :

$$\mathrm{THD} = \sqrt{\frac{V_{f2}^{2} + V_{f3}^{2} + \ldots + V_{fn}^{2}}{V_{f1}^{2}}}$$

where Vf1 is the fundamental amplitude, Vf2 is the second harmonic amplitude, etc

THD performance degrades with increasing frequency because the effects of jitter get worse and because the input circuitry becomes slew limited

ADC Characteristics-SFDR



Spurious Free Dynamic Range (SFDR) is the difference between the value of the desired output signal and the value of the highest amplitude output frequency that is not present in the input, expressed in dB.

ADC Characteristics-(SINAD- relation to SNR)

SINAD is defined as the RMS value of an input sine wave to the RMS value of the noise of the converter (from DC to the Nyquist frequency, including harmonic [total harmonic distortion] content). Harmonics occur at multiples of the input frequency .

$$\text{SINAD} = -20 * \text{Log}\sqrt{10^{\frac{-\text{SNR}}{10}} + 10^{\frac{\text{THD}}{10}}}$$

SNR is similar to SINAD, except that it does not include the harmonic content. Thus, the SNR should always be better than the SINAD. Both SINAD and SNR are typically expressed in dB.

$$\text{SINAD} = [6.02\,(N) + 1.76]\ (\text{dB})$$

where N is the number of bits. For an ideal 12-bit converter, the SINAD is 74dB. Should this equation be
rewritten in terms of N, it would reveal how many bits of information are obtained as a function of the
RMS noise

$$N = (\text{SINAD} - 1.76)/6.02$$

This equation is the alternative definition for effective number of bits, or ENOB.

ADC Characteristics-Input Dynamic Range-ADC Figure of Merit

Input Dynamic Range (sometimes just called Dynamic Range) is the ratio of the largest to the smallest signal that can be resolved.
The largest output code, of course, is 2exp(n) -1 and the smallest output code, greater than 0, is 1. Dynamic range in dB, then, is:

$$20 * \mathrm{Log}((2^n -1)/1) = 20 * \mathrm{Log}(2^n -1)$$

A popular Figure-of-Merit (FOM) used to compare different ADCs is

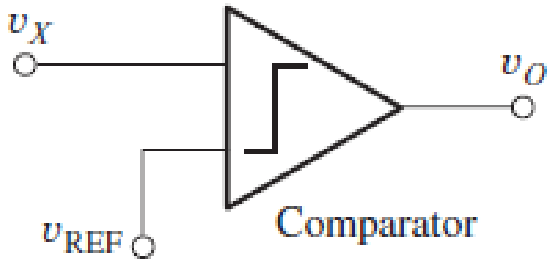
$$FOM = \frac{Power}{(2^{ENOB})(f_s)} \ (\mathrm{pJ/step})$$

where fs is the sampling rate in Nyquist-rate ADCs. This figure of merit is commonly used to compare published reports as it is based on easily measured quantities, and calculates a value that has meaningful units (i.e. energy required per conversion step)
-Lower FOM means a better ADC

In general similar FOMs can be achieved with different ADC topologies, however it is noted that ADCs with lower resolutions tend to be able to achieve better FOMs

ADC Techinques

General Concept



Block diagram representation for an A/D converter.

Basic conversion scheme for a number of analog-to-digital converters.

The unknown input voltage *υχ is connected to one input of an analog* ***comparator, and a time-dependent*** reference voltage *υREF is connected to the other input of the comparator.*

If input voltage υχ exceeds input *υREF, then the output voltage will be high, corresponding to a logic 1.*

If input υχ is less than υREF, then the output voltage will be low, corresponding to a logic 0.

In performing a conversion, the reference voltage is varied until the unknown input is determined within the quantization error of the converter.

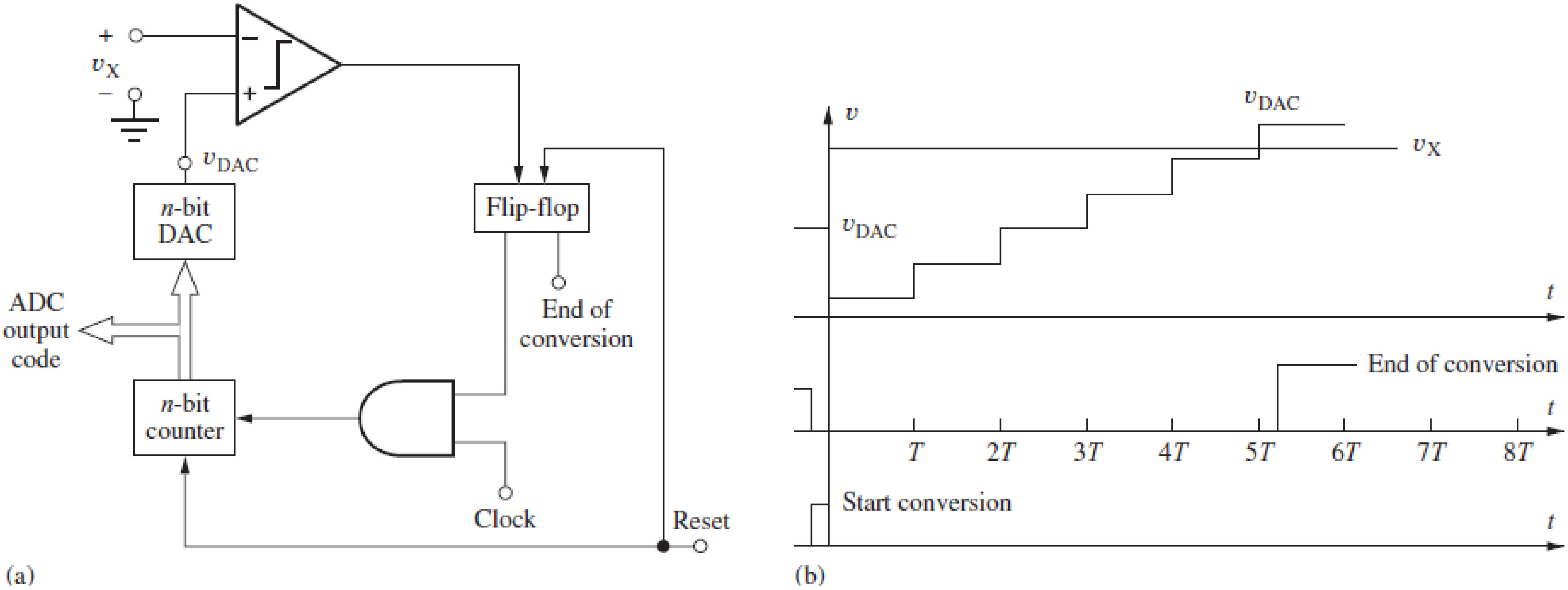
Ideally, the logic of the A/D converter will choose a set of binary coefficients *bi so that the difference between the unknown input voltage υχ and the* final quantized value is less than or equal to 0.5 LSB. In other words, the *bi will be selected so that*

$$\left| v_X - V_{FS} \sum_{i=1}^{n} b_i 2^{-i} \right| < \frac{V_{FS}}{2^{n+1}}$$

The basic difference among the operations of various converters is the strategy that is used to vary the reference signal *VREF to determine the set of binary coefficients {bi , i = 1 . . . n}.*

Counting Type Converters

Counting Converter



(a) Block diagram of the counting ADC. (b) Timing diagram.

A/D conversion begins when a pulse resets the flip-flop and the counter output to zero. Each successive clock pulse increments the counter; the DAC output looks like a staircase during the conversion.

When the output of the DAC exceeds the unknown input, the comparator output changes state, sets the flip-flop, and prevents any further clock pulses from reaching the counter. The change of state of the comparator output indicates that the conversion is complete. At this time, the contents of the binary counter represent the converted value of the input signal

Counting Converter

Discussion: Features-Advantages –Disadvantages

- First, the length of the conversion cycle is variable and proportional to the unknown input voltage *υχ*
- *The maximum* ***conversion time T$_T$*** occurs for a full-scale input signal and corresponds to 2expn *clock periods or*

$$T_T \leq \frac{2^n}{f_C} = 2^n T_C$$ where *fc = 1/Tc is the clock frequency.*

Also, the example in previous Fig (b) shows the case for an input that is constant during the conversion period. If the input varies, the binary output will be an accurate representation of the value of the input signal at the instant the comparator changes state.

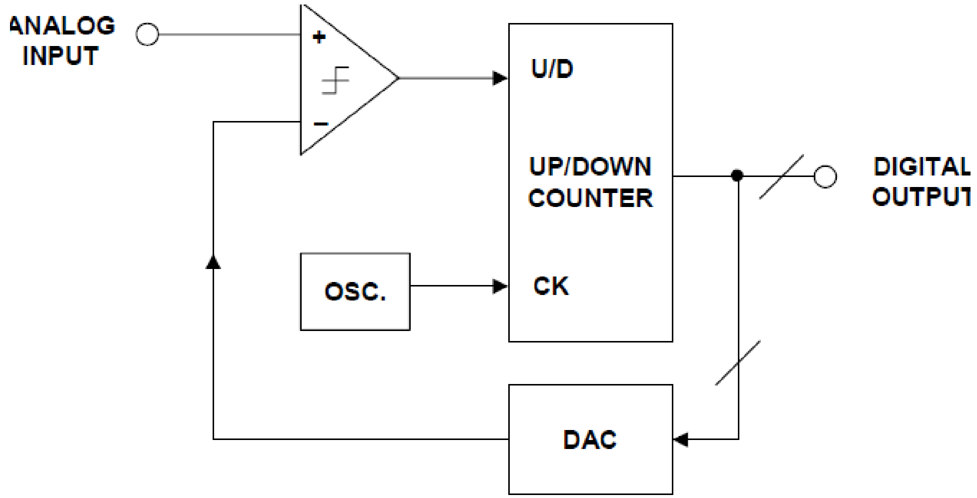
The advantage of the counting A/D converter is that it requires a minimum amount of hardware and is inexpensive to implement. Some of the least expensive A/D converters have used this technique.

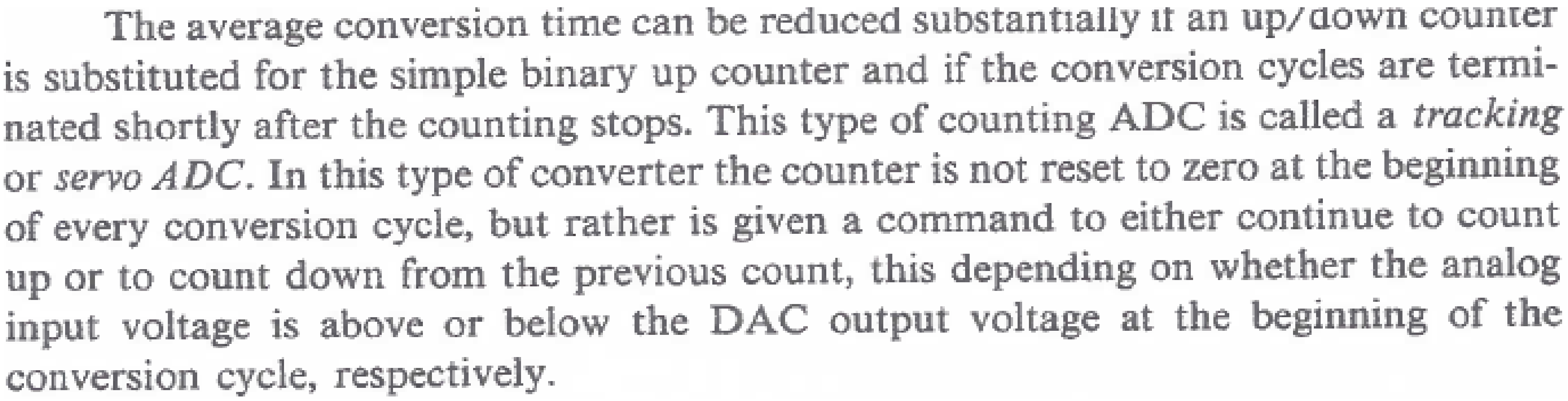
The main disadvantage is the relatively low conversion rate for a given D/A converter speed.

An *n-bit converter requires 2expn clock periods for its longest conversion.*

For a counting ADC using a 12-bit DAC and a 2-MHz clock frequency the maximum conversion time is 2.05 ms

Tracking or servo Converter




The average conversion time can be reduced substantially if an up/down counter is substituted for the simple binary up counter and if the conversion cycles are terminated shortly after the counting stops. This type of counting ADC is called a *tracking* or *servo ADC*. In this type of converter the counter is not reset to zero at the beginning of every conversion cycle, but rather is given a command to either continue to count up or to count down from the previous count, this depending on whether the analog input voltage is above or below the DAC output voltage at the beginning of the conversion cycle, respectively.

The tracking ADC architecture shown in Figure continually compares the input signal with a reconstructed representation of the input signal.

The up/down counter is controlled by the comparator output. If the analog input exceeds the DAC output, the counter counts up until they are equal. If the DAC output exceeds the analog input, the counter counts down until they are equal

Tracking or servo Converter

Discussion

It is evident that if the analog input changes slowly, the counter will follow, and the digital output will remain close to its correct value:

If the analog input suddenly undergoes a large step change, it will be many hundreds or thousands of clock cycles before the output is again valid.

The tracking ADC therefore responds quickly to slowly changing signals, but slowly to a quickly changing one.

In order that the ADC be able to track the voltage input its change rate should be < or = to converter change rate , i.e.:

$$\frac{dV}{dt} = \frac{Vfs \cdot Fclk}{2^N}$$

Tracking ADCs are not very common. Their slow step response makes them unsuitable for many applications, but they do have one asset: their output is *continuously available*

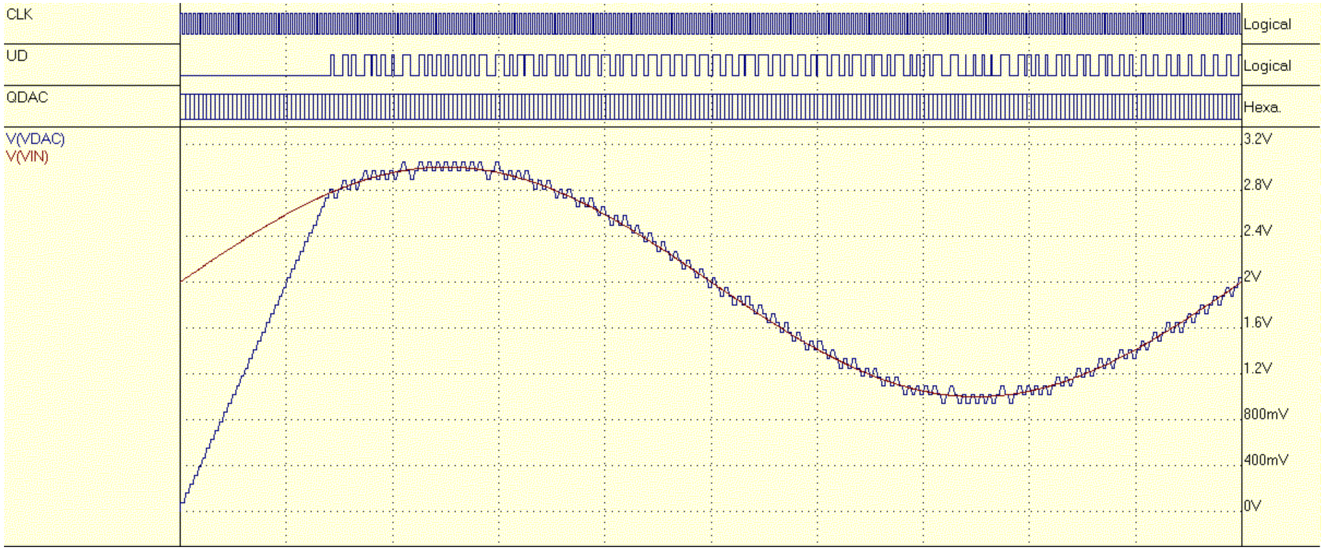
Most ADCs perform conversions: i.e., on receipt of a "start convert" command they perform a conversion and, after a delay, a result becomes available. In a tracking ADC though providing that the analog input changes slowly, its output is always available.

Another valuable characteristic of tracking ADCs is that a fast transient on the analog input causes the output to change only one count. This is very useful in noisy environments.

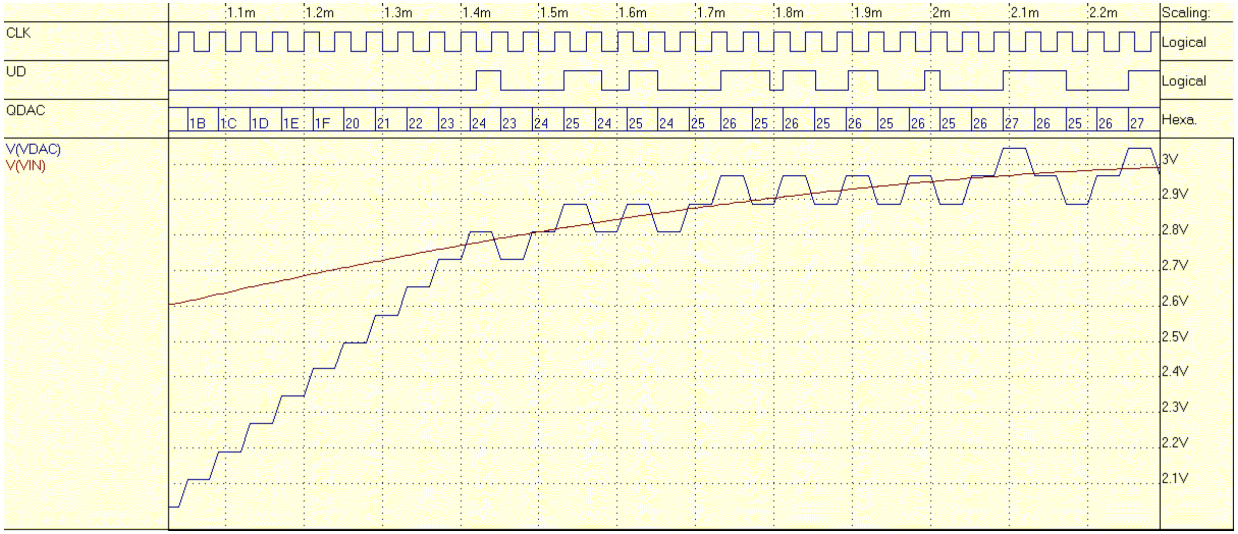
In general in counting-type converters their accuracy is a function of the offset voltage and voltage gain of he comparator and of the DAC accuracy , which is often the dominant limiting factor

Tracking or servo Converter

Diagrams showing the covergence of the ADC and the tracking of the input signal



Second Dagram below focused on the detail of the tracking of the converter to the input signal



Successive Approximation Converters

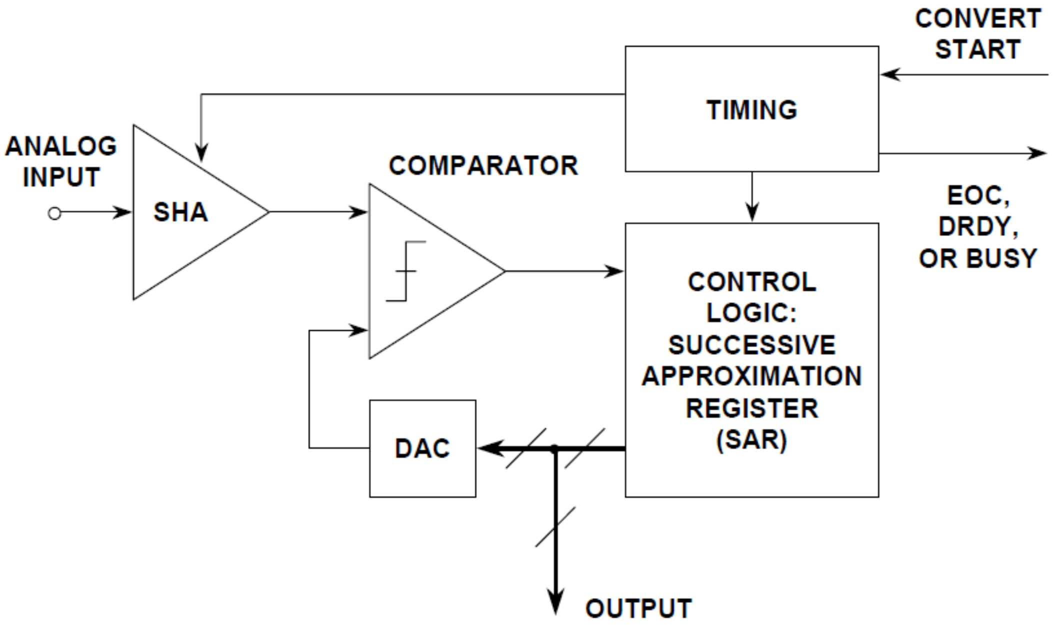
Successive Approximation Converter

Basic Concept

The **successive approximation converter uses a much more efficient strategy for varying the reference** input to the comparator, one that results in a converter requiring only *n clock periods to* complete an *n-bit conversion*

A “binary search” is used to determine the best approximation to analog input signal

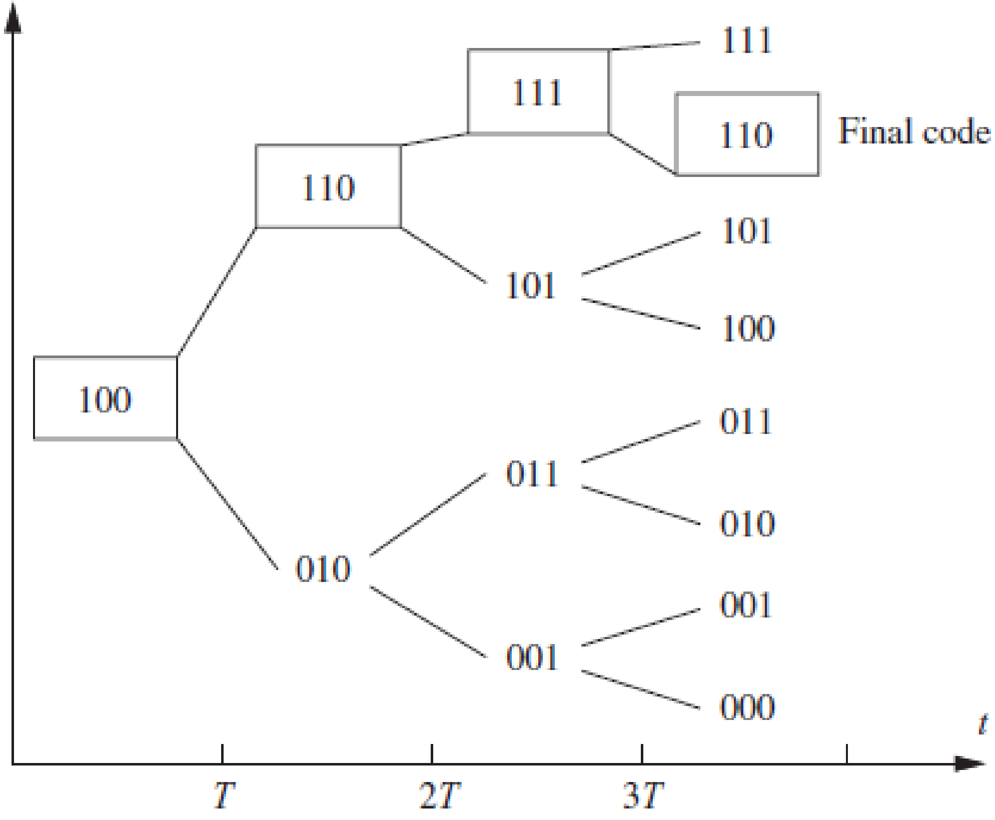
Successive Approximation Converter



On the assertion of the CONVERT START command, the sample-and-hold (SHA) is placed in the *hold mode, and all the bits of the successive* approximation register (SAR) are reset to "0" except the MSB which is set to "1".
The SAR output drives the internal DAC. If the DAC output is greater than the analog input, this bit in the SAR is reset, otherwise it is left set.
The next most significant bit is then set to "1". If the DAC output is greater than the analog input, this bit in the SAR is reset, otherwise it is left set. The process is repeated with each bit in turn.
When all the bits have been set, tested, and reset or not as appropriate, the contents of the SAR correspond to the value of the analog input, and the conversion is complete.

Successive Approximation Converter

A 3-bit Successive Approximation ADC example



Code sequences for a 3-bit successive approximation AD

Successive Approximation Converter

Discussion: Advantages and Disadvantages

An N-bit conversion takes N steps. Fast conversion rates are possible with a successive approximation ADC. This conversion technique is very popular and used in many 8 to 16-bit converters.

The successive approximation ADC has been the mainstay of data acquisition for many years. Recent design improvements have extended the sampling frequency of these ADCs into the megahertz region.

The resolution of these ADCs can be extended to 18-bits on CMOS processes

The primary factors limiting the speed of this ADC are the time required for the D/A converter output to settle within a fraction of an LSB of VFS and the time required for the comparator to respond to input signals that may differ by very small amounts.

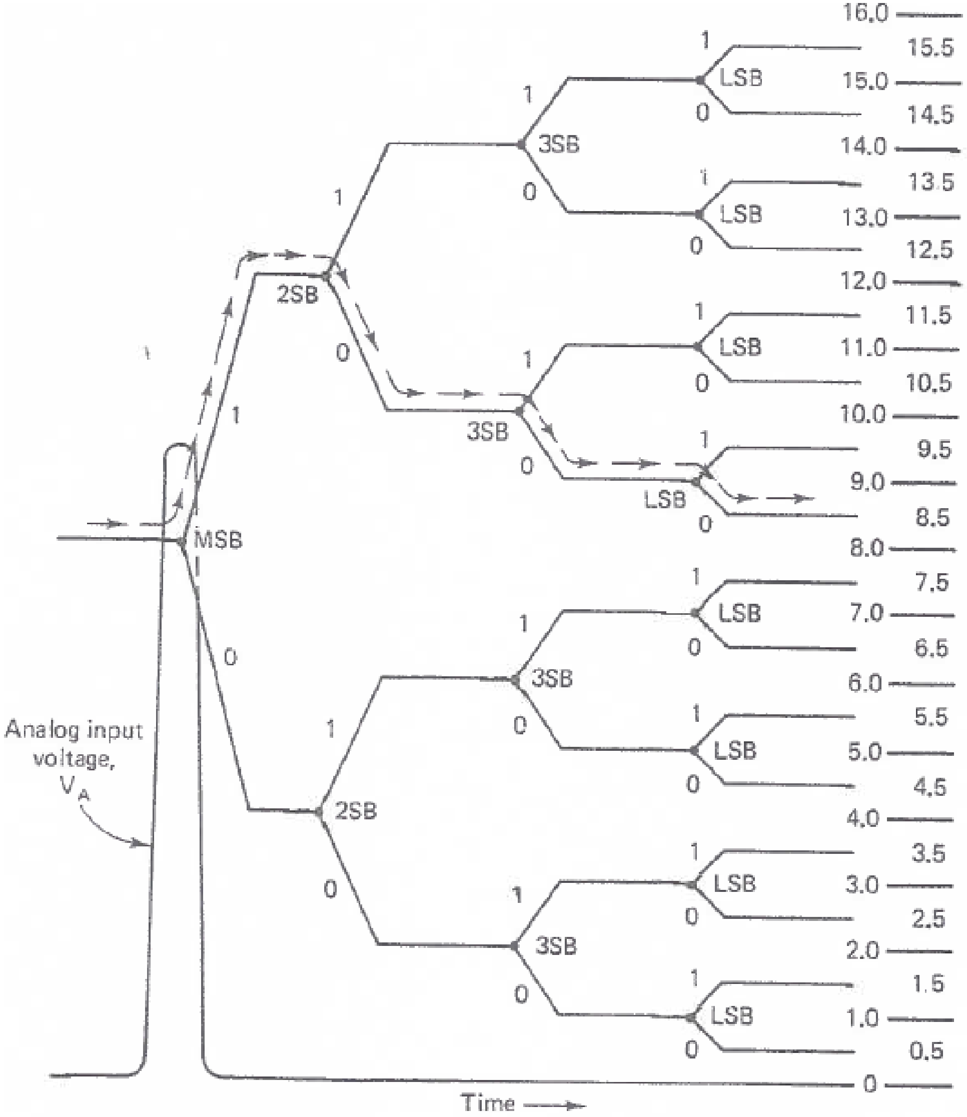
For Example it would seem on superficial examination that a 16-bit converter would have twice the conversion time of an 8-bit one, but this is not the case.

In an 8-bit converter, the DAC must settle to 8-bit accuracy before the bit decision is made, whereas in a 16-bit converter, it must settle to 16-bit accuracy, which takes a lot longer

This is mainly due to the fact that the comparator gets much slower when the differences in the voltages to be compared are getting smaller

In practice, 8-bit successive approximation ADCs can convert in a few hundred nanoseconds, while 16-bit ones will generally take several microseconds

Successive Approximation Converter



Susceptible to input variations and spikes. Figure shows what happens when input voltage = 0 and a large voltage spike is faced. Instead of 0000 the ADC output resides to 1000!

Successive Approximation Converter

Advantages and Disadvantages - Maximum frequency of input signal tolerable

Thus far, it has been tacitly assumed that the input remains constant during the full conversion period.

A slowly varying input signal is acceptable as long as it does not change by more than 0.5 LSB *(VFS/2 exp (n+1)) during the conversion time* $T_T = n/ f_C = nT_C$.

The frequency of a sinusoidal input signal with a peak-to-peak amplitude equal to the full-scale voltage of the converter must satisfy the following inequality:

$$T_T\left\{\max\left[\frac{d}{dt}(V_{FS}\sin\omega_o t)\right]\right\} \leq \frac{V_{FS}}{2^{n+1}} \qquad \text{or} \qquad \frac{n}{f_C}(V_{FS}\omega_o) \leq \frac{V_{FS}}{2^{n+1}} \qquad =>$$

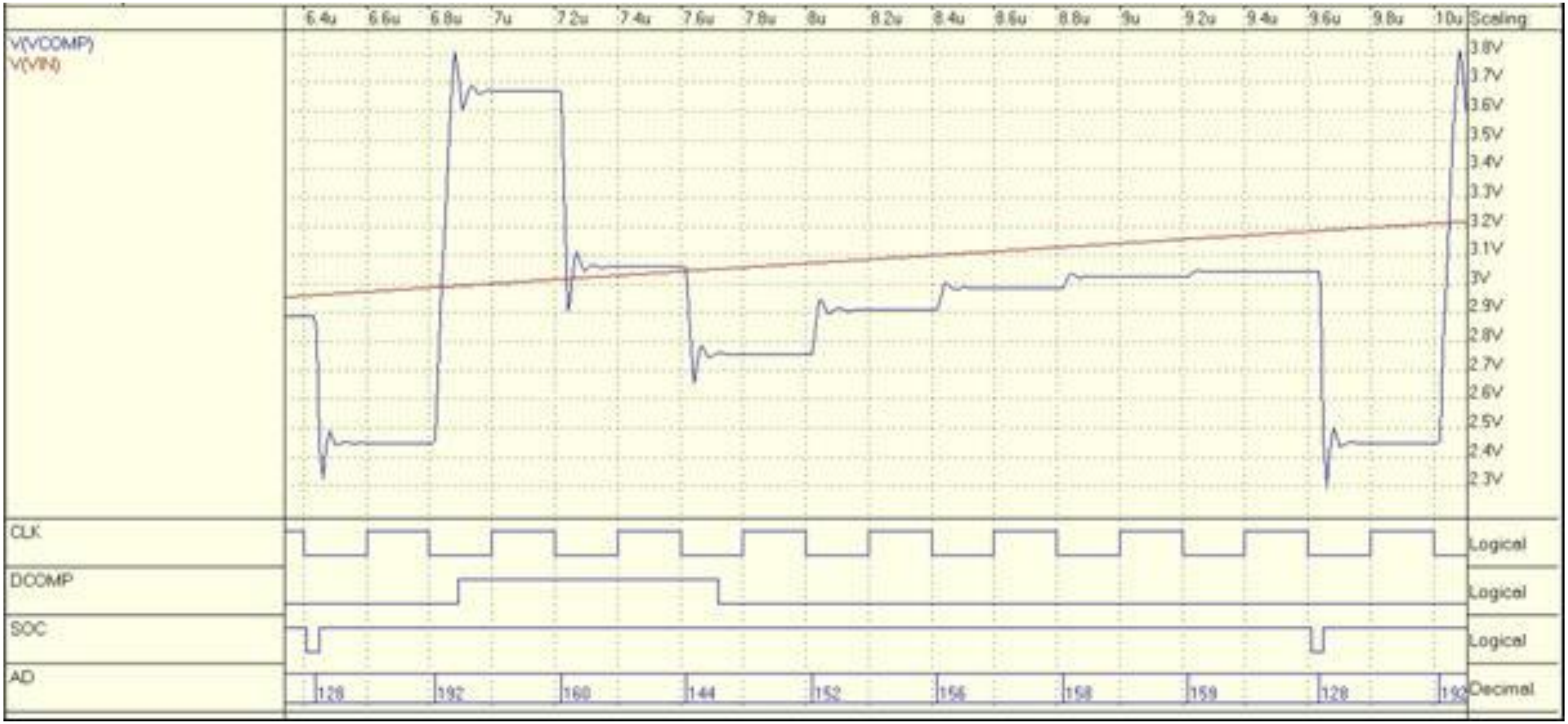
$$f_O \leq \frac{f_C}{2^{n+2}n\pi}$$

Example: For a 12-bit converter using a 1-MHz clock frequency, f_O *must be less than 1.62 Hz. If the input* changes by more than 0.5 LSB during the conversion process, the digital output of the converter does not bear a precise relation to the value of the unknown input voltage υχ *.*

To avoid this frequency limitation, a high-speed sample-and-hold circuit that samples the signal amplitude and then holds its value constant is usually used ahead of successive approximation ADCs.

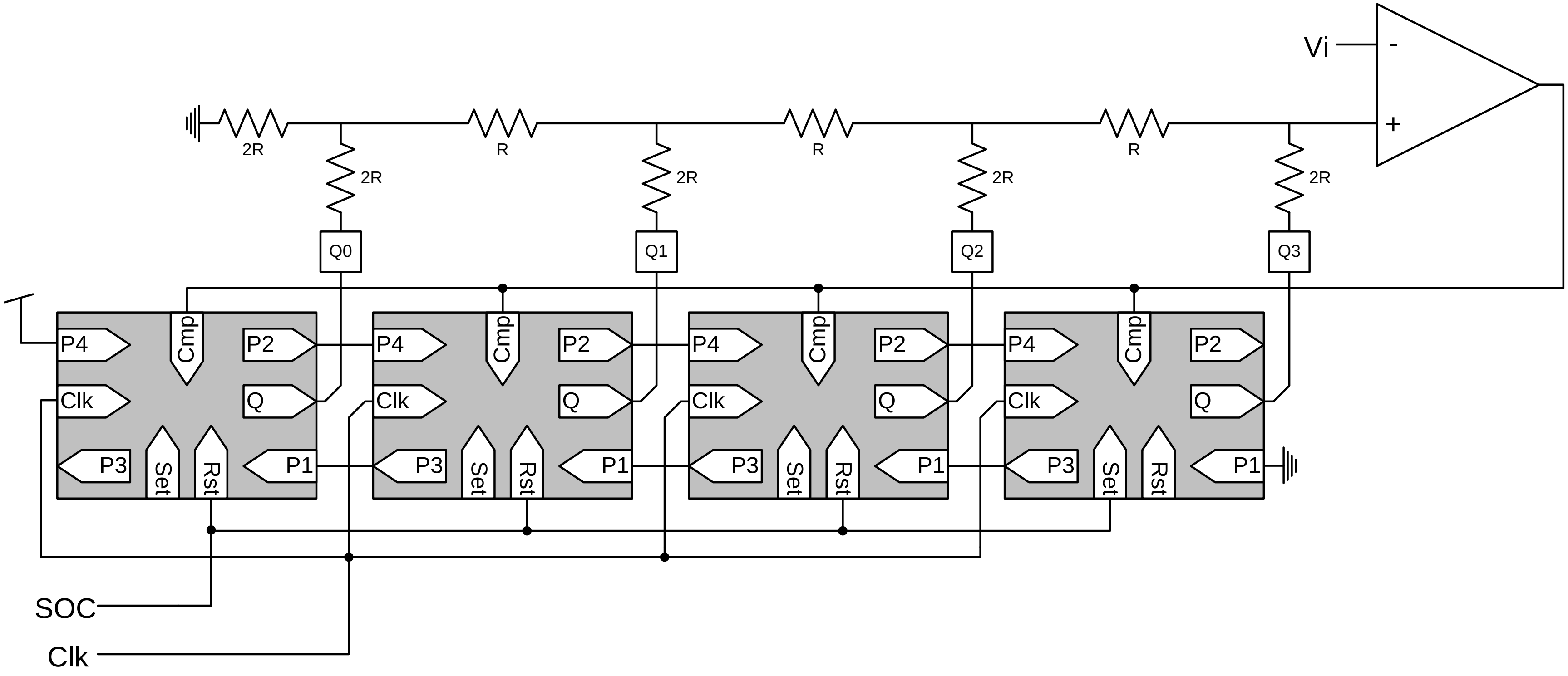
Successive Approximation Converter

A/D Behaviour in presence of analog input with large variation rate



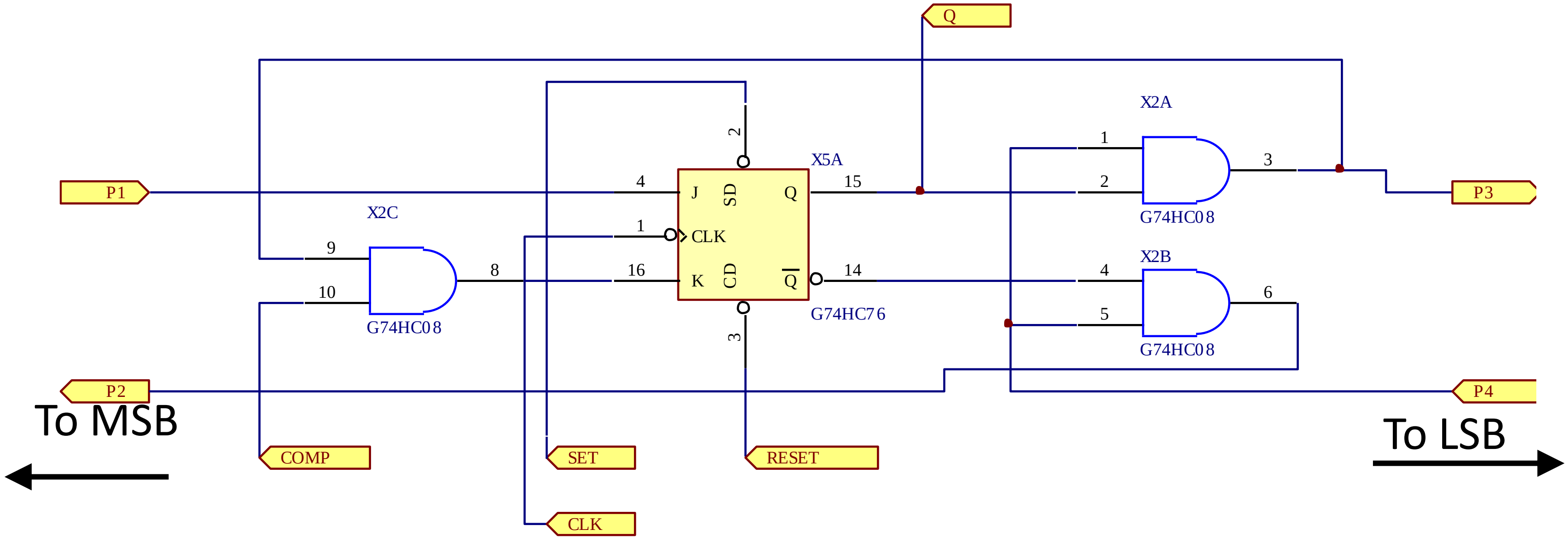
Successive Approximation Converter

This converter is greatly based on the SUCCESSIVE APPROXIMATION REGISTER block consisting of n cells



Successive Approximation Converter

Δομή ενός SAR Κυττάρου



Αρχές Λειτουργίας: Κάθε κύτταρο ακολουθεί τους τρεις παρακάτω απλούς κανόνες.

Α) *Κάθε κύτταρο πηγαίνει στην κατάσταση High όταν το σήμα P1 είναι High:*

- *όλα τα επόμενα λιγότερο σημαντικά κύτταρα είναι Low*
- *<u>και</u> το αμέσως προηγούμενο έχει τεθεί σε High.*

Μόνο το πρώτο κύτταρο πηγαίνει στην κατάσταση High με τον παλμό SOC

Β) *Κάθε κύτταρο (συμπεριλαμβανομένου και του πρώτου) πηγαίνουν στην κατάσταση Low όταν:*

- *το κύτταρο είναι σε κατάσταση High*
- *και το σήμα VCOMP είναι High => η τάση εξόδου του DAC είναι μεγαλύτερη από την τάση εισόδου*
- *και όλα τα επόμενα λιγότερο σημαντικά κύτταρα είναι Low*

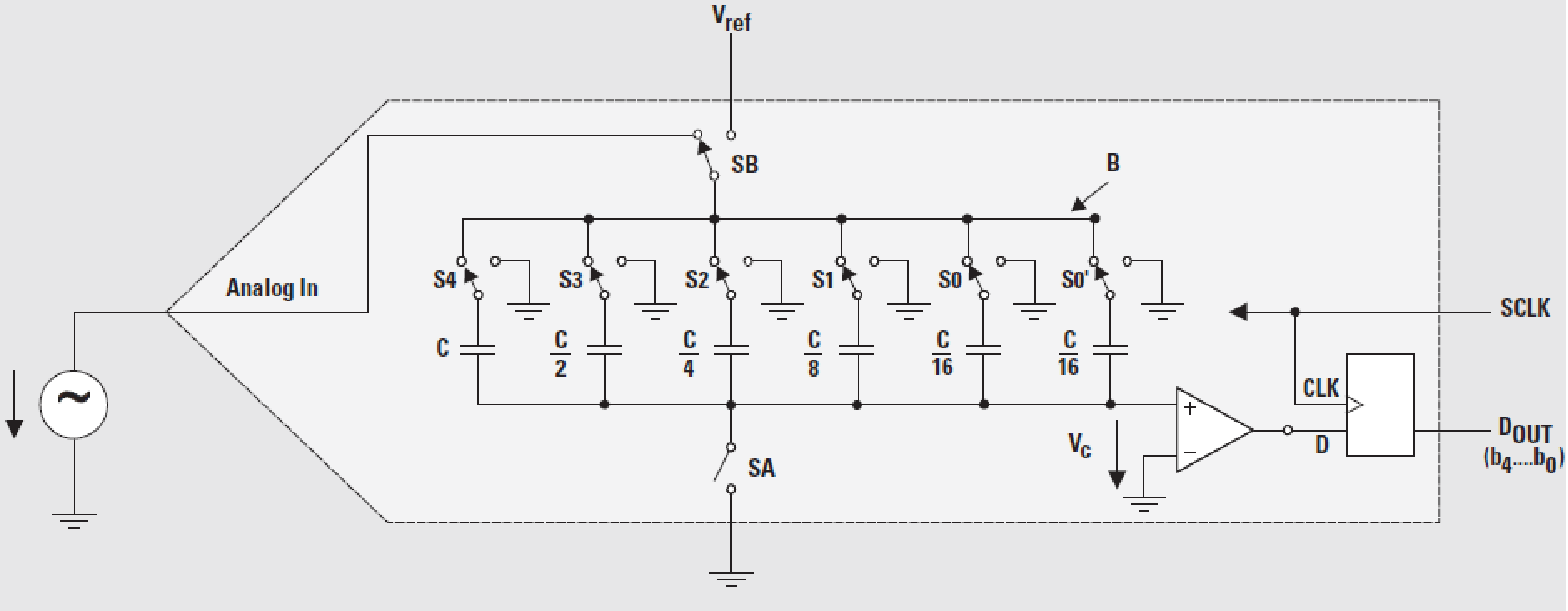
Γ) *Όταν έρχεται ο παλμός SOC τότε το πρώτο κύτταρο τίθεται σε High και όλα τα άλλα σε Low*

Αρχές Λειτουργίας και I/Os ενός κυττάρου SAR

- Η είσοδος **CLK** ενεργοποιείται στην πίπτουσα παρυφή και τροποποιεί κατάλληλα την έξοδο του FF ανάλογα με τις καταστάσεις των J και K.
- Η είσοδος **SET** χρησιμοποιείται για να θέσει το MSBit FF ασύγχρονα στην αρχικοποίηση του SAR.
- Η είσοδος **RESET** χρησιμοποιείται για να καθαρίσει τα υπόλοιπα LSBits FFs ασύγχρονα στην αρχικοποίηση του SAR. Έτσι κατά την αρχικοποίηση ο SAR τίθεται στην τιμή 2^{N-1} N = αριθμός bit
- Η είσοδος **COMP** πληροφορεί το κύτταρο για την στάθμη της τάσης εισόδου του A/D σε σχέση με την τάση του D/A Ladder. Η είσοδος COMP είναι High όταν η τάση εξόδου του Ladder > από την τάση εισόδου. Η είσοδος COMP θέτει την είσοδο K του FF σε High, υποχρεώνοντάς το να καθαριστεί όταν η είσοδος COMP είναι High <u>και</u> το FF είναι σε κατάσταση High <u>και</u> τα περισσότερο σημαντικά FF είναι σε κατάσταση Low (έξοδος P3, είσοδος P4).
- Η είσοδος **P1** οδηγείται από την έξοδο P3 του προηγούμενου (MS) κύτταρου. Το σήμα P1 είναι High, αναγκάζοντας το FF να τεθεί σε High στην επόμενη παρυφή του CLK, όταν το προηγούμενο (περισσότερο σημαντικό FF) είναι High και τα επόμενα (λιγότερο σημαντικά FFs) είναι Low.
- Η είσοδος **P4** οδηγείται από την έξοδο P2 του επόμενου (LS) κύτταρου.
- Η έξοδος **P3** είναι High όταν το FF του κύτταρου είναι High και όλα τα λιγότερο σημαντικά FF είναι Low.
- Όταν η έξοδος P3 είναι High, τότε το αμέσως λιγότερο σημαντικό FF θα τεθεί στον επόμενο κύκλο του CLK και το τρέχον FF θα καθαριστεί εφ' όσον η είσοδος COMP είναι High, δηλαδή η αναλογική έξοδος του D/A > από την αναλογική είσοδο του A/D.

Charge –Redistribution Converter

Five-bit ADC example based on charge redistribution



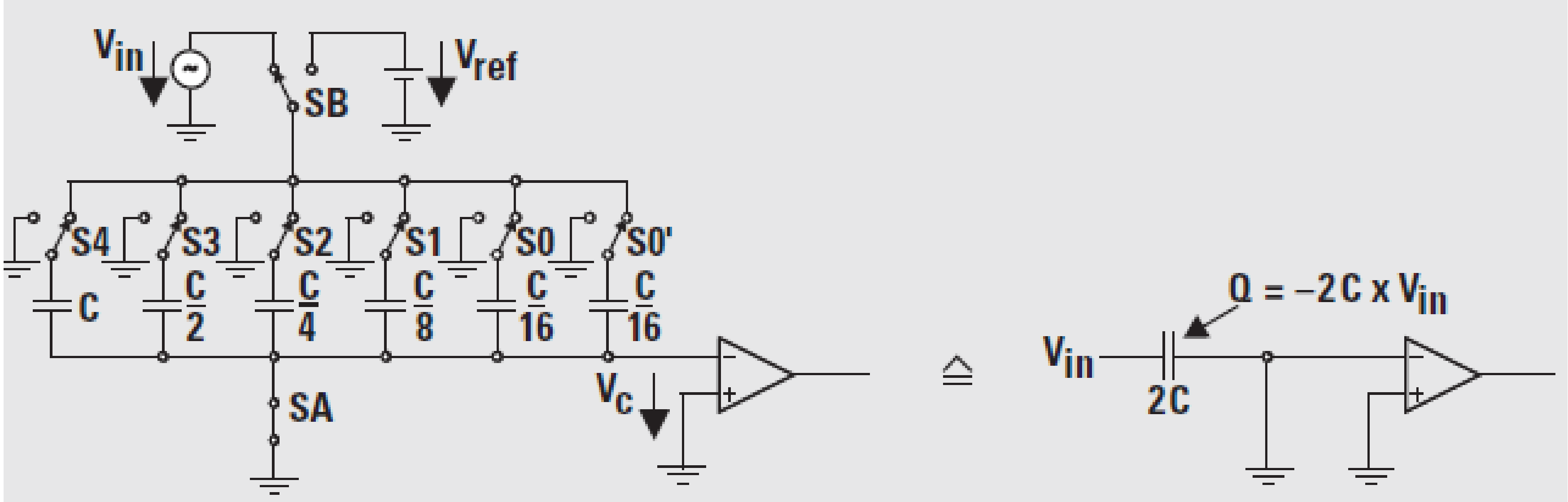
All capacitors have binary weighted values, i.e., C, C/2, C/4,….C/2exp(n–1). The last two capacitors having the value C/2exp(n–1) are connected so that total capacitance =2C. MOS-transistors are used to implement the required n+3 switches, and the voltage comparator provides the appropriate steering of the switches via auxiliary logic circuitry.

The conversion process is performed in three steps: The sample mode, the hold mode, and the redistribution mode (in which the actual conversion is performed).

Insensitivity to stray capacitances makes this technique a reasonably accurate method capable of implementing A/D converters with as many as 10 bits

Charge –Redistribution Converter

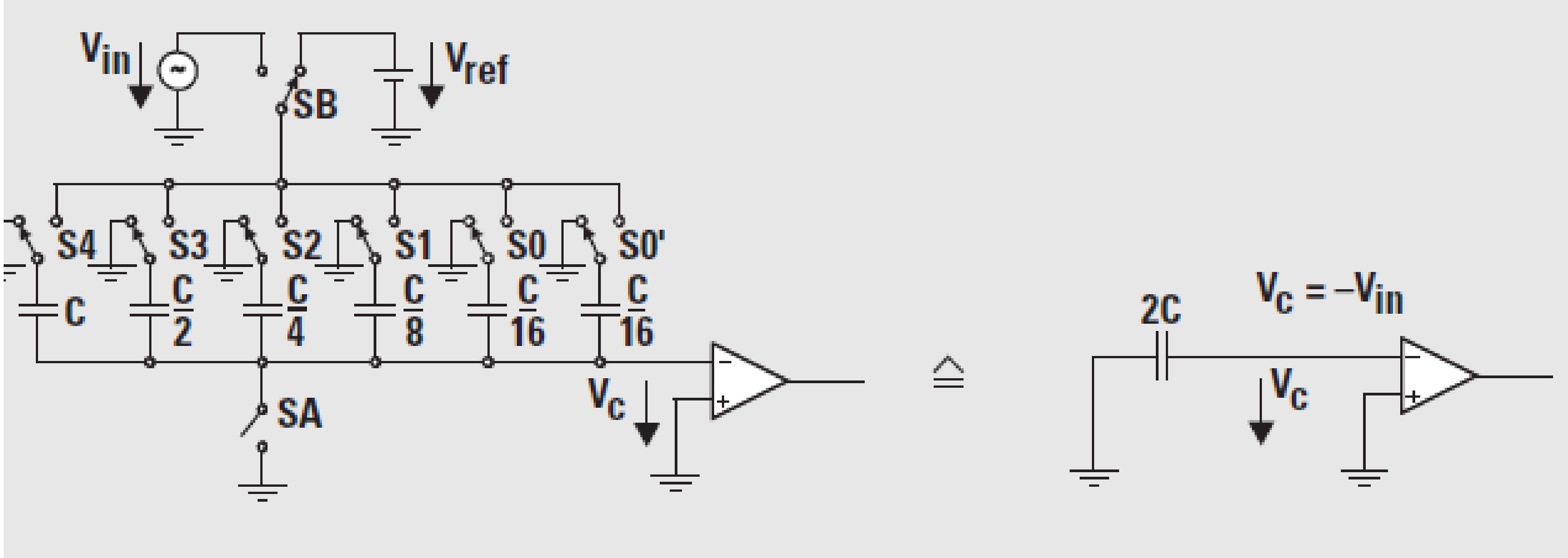
Sample mode



In the sampling mode, switch SA is closed and SB is switched to the input voltage Vin.
The remaining switches are turned to the common bus B.
Due to charging, a total charge of Qin = –2C x Vin is stored on the capacitors.

Charge –Redistribution Converter

Hold mode

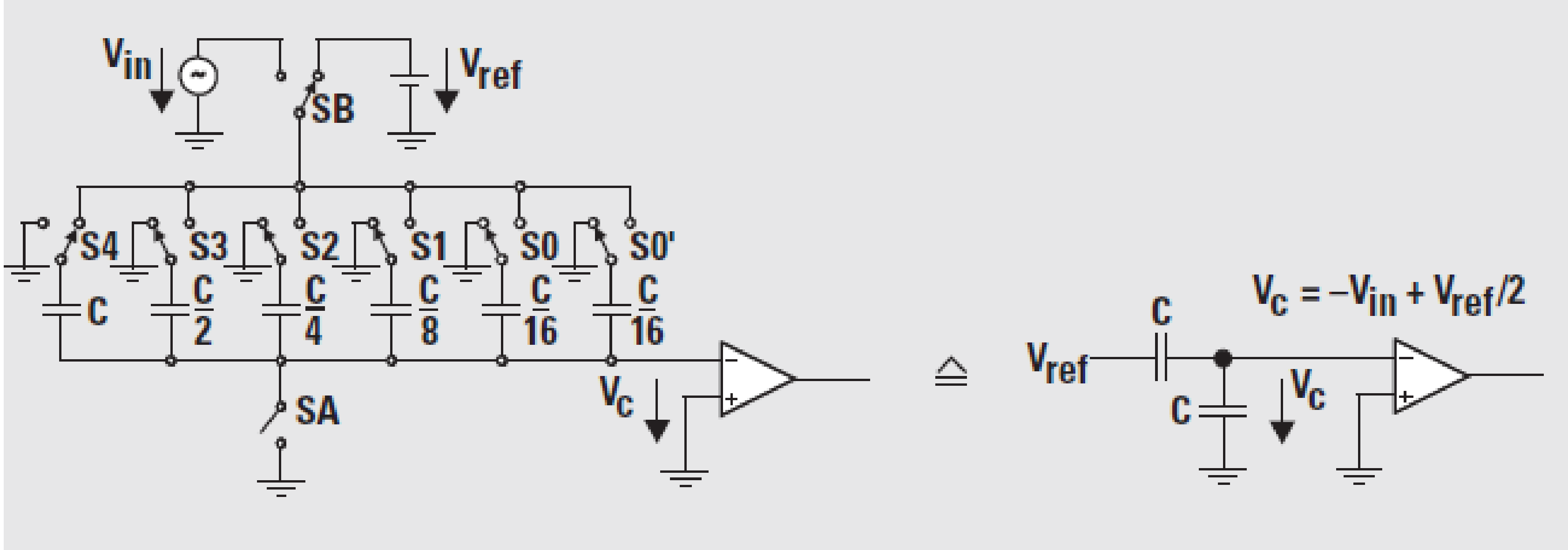


During the hold mode, switch SA is opened while the switches S4….S0’ are connected to ground

The result is that a voltage of Vc = –Vin is applied to the comparator input. This means that the circuit already has a built-in sample-and-hold element

Charge –Redistribution Converter

Redistribution mode-Conversion Step 1 determines the MSB (bit 4)

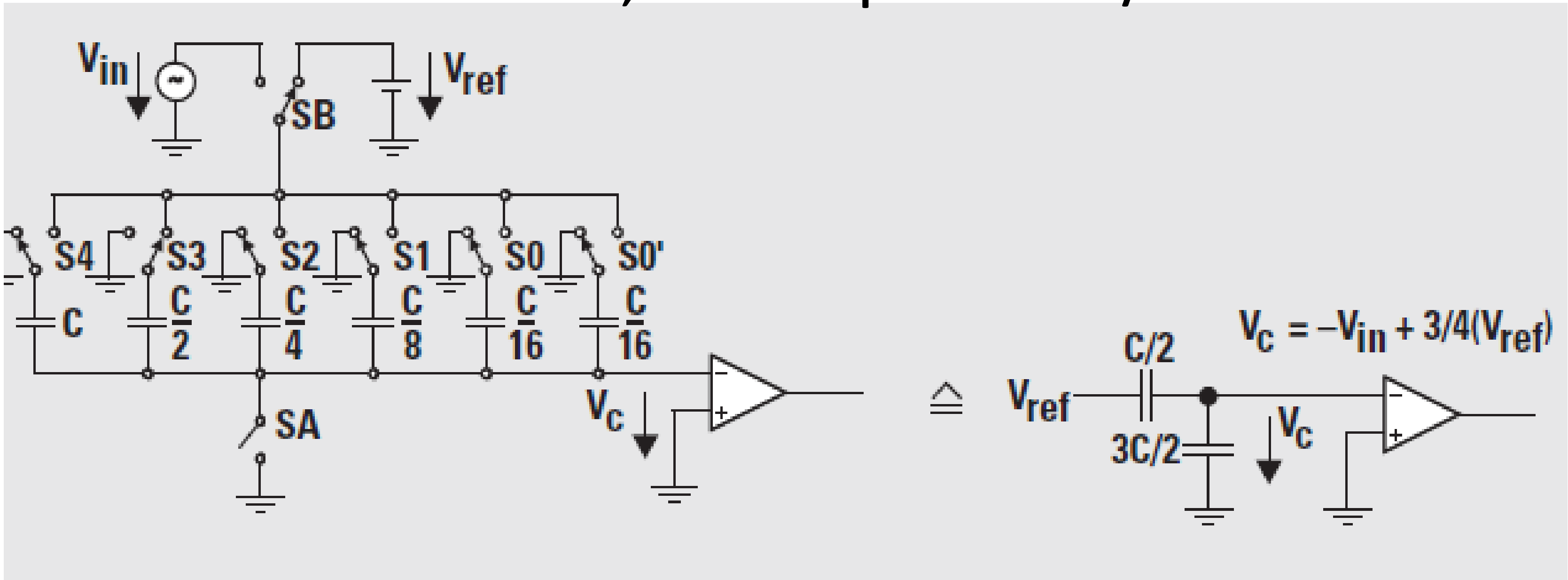


The actual conversion is performed by the redistribution mode.
The first conversion step, shown in Figure, connects C (the largest capacitor) via switch S4 to the reference voltage Vref, which corresponds to the full-scale range (FSR) of the ADC.

Capacitor C forms a 1:1 capacitance divider with the remaining capacitors connected to ground. The comparator input voltage becomes Vc = –Vin + Vref /2.
If Vin > Vref /2, then Vc < 0, and the comparator output goes high, providing the most significant bit MSB (bit 4) = 1.
On the other hand, if Vin < Vref /2, then Vc > 0, and bit 4 = 0.

Charge –Redistribution Converter

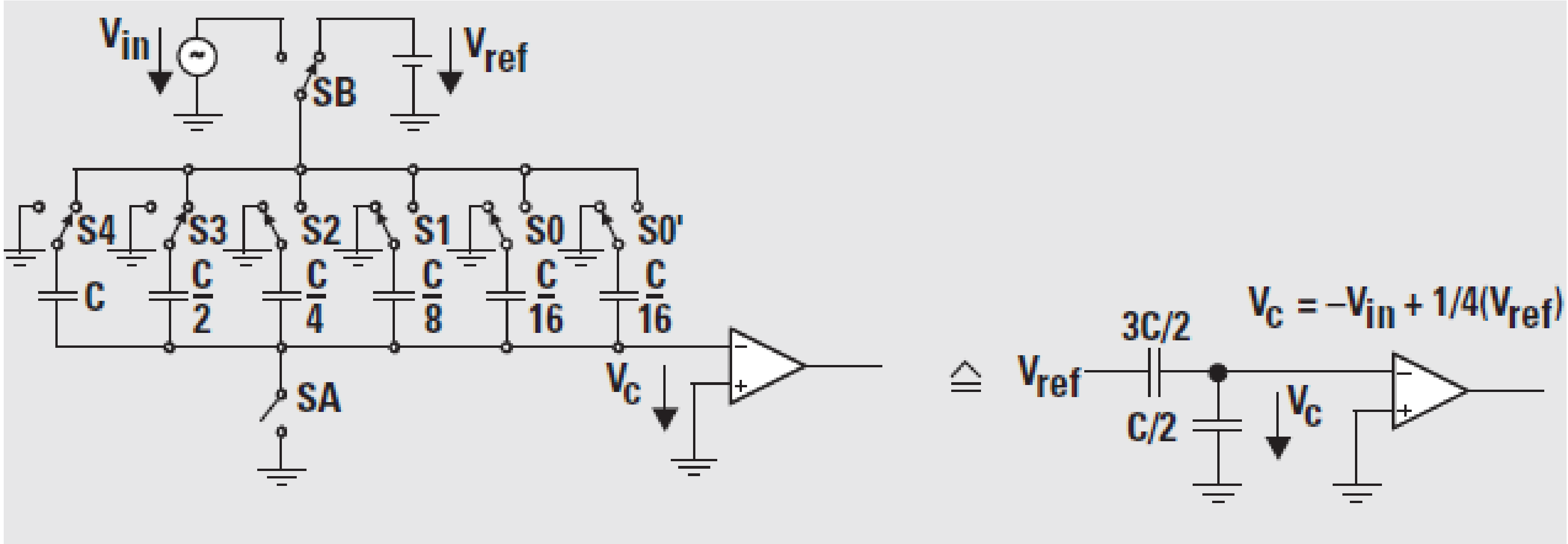
Redistribution mode- If bit 4 = 1, Vin is compared with 3/4 Vref



The second conversion step connects C/2 to Vref .
If the first conversion step resulted in bit 4 = 1, switch S4 is turned to ground again to discharge C as shown in Figure
Thus Vin is compared with 3/4 Vref since Vc= -Vin + 3/4 Vref through the voltage dividers formed by the capacitors configuration.

Charge –Redistribution Converter

Redistribution mode- If bit 4 = 0, Vin is compared with 1/4 Vref

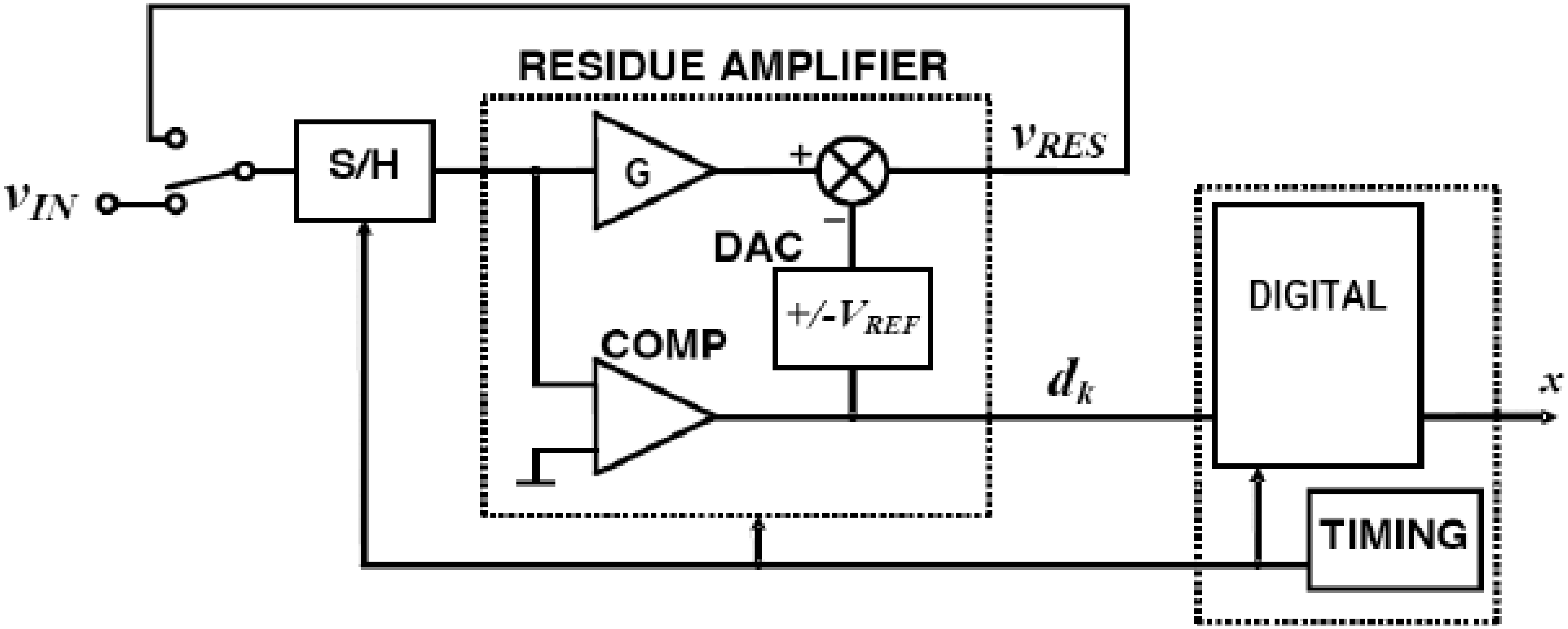


The second conversion step connects C/2 to Vref .
If the first conversion step resulted in bit 4 = 0, switch S4 remains connected to Vref. In this case Vin is compared with 1/4 Vref since Vc= -Vin + 1/4 Vref through the voltage dividers formed by the capacitors configuration
Thus depending on the value of bit 4 the comparator input voltage can be written as : Vc = -Vin + bit 4 x Vref /2 + Vref /4.

This process continues until all bits are generated, with the final conversion step being performed at a comparator input voltage of Vc = –Vin + bit 4 x Vref /2 + bit 3 x Vref /4 + bit 2 x Vref /8 + bit 1 x Vref /16 + bit 0 x Vref /32.

Algorithmic (Cyclic) ADC

A Cyclic converter, also known as an Algorithmic converter, is similar in operation to the successive approximation converter, where in the case of the Cyclic ADC, the reference voltage is not altered. Instead, the error (or residue) of the amplifier is doubled



Algorithmic (Cyclic) ADC

Basic Mechanism

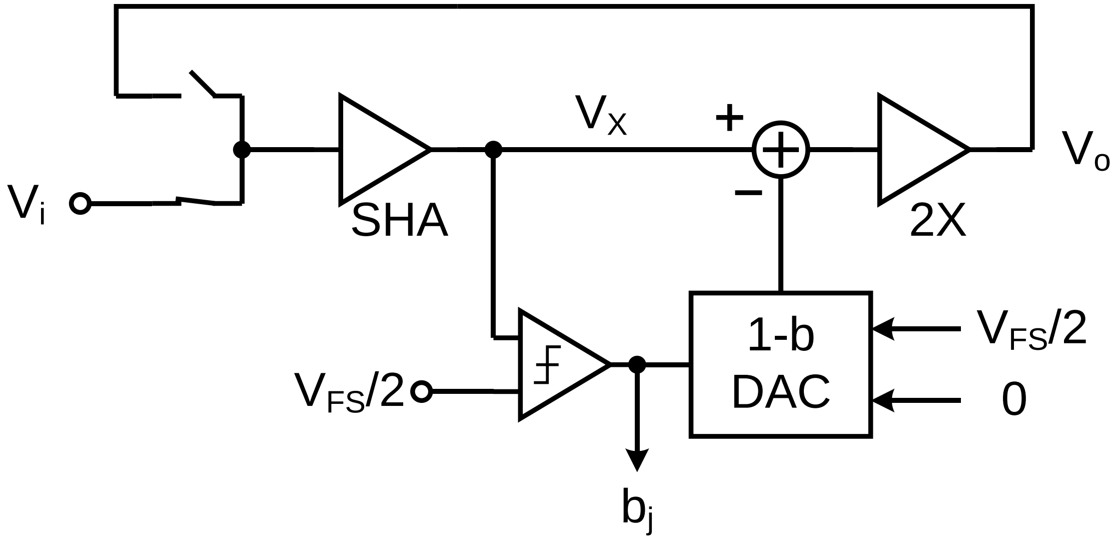
The operation of the cyclic converter functions in the following manner: First, the input voltage is sampled by the S-H block. That value is then compared to a threshold voltage, upon which a digital decision is made, determining a bit value in the final sequence of the number sampled.

A reference voltage is generated by a 1-bit digital-to-analog converter which is dictated by the digital decision previously made. At the same time, the input value is amplified by a factor of two (ideally).

The amplified value is then summed to a reference voltage +/- VREF, leaving a residue voltage. The residue voltage then becomes the input of the residue amplifier.

This cycle is repeated enough (N) times to achieve the desired resolution, earning the device its name. The sequence of decisions corresponds to the output value of the ADC.

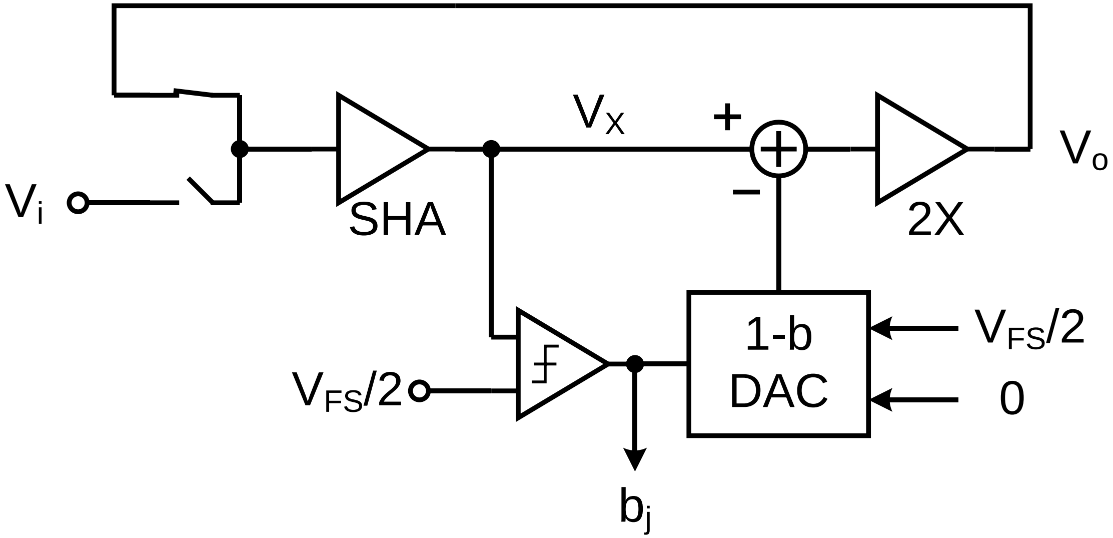
Algorithmic (Cyclic) ADC



<u>Sample mode</u>

- Input is sampled first, then circulates in the loop for N clock cycles
- Conversion takes N cycles with one bit resolved in each T_{clk}

Modified Binary Search



Conversion mode

- If $V_X < V_{FS}/2$, then $b_j = 0$, and $V_o = 2*V_X$
- If $V_X > V_{FS}/2$, then $b_j = 1$, and $V_o = 2*(V_X-V_{FS}/2)$
- V_o is called conversion “**residue**”

Algorithmic (Cyclic) ADC

Understanding the Residue Amplifier –Analytical Calculation of ADC output

A major part of the cyclic ADC is the residue amplifier. Therefore, in order to better comprehend the operation of the ADC, we can take a mathematical approach to explain this concept. The equation below shows the relationship between the residue amplifier’s input and output:

$$v_{res_{out}} = G \cdot v_{res_{in}} - d \cdot V_{ref} \qquad \text{Eq. 6}$$

where *G is the gain of the amplifier and d is the digital decision*

$$v_{res_{out}}(N) = \left[G^N \cdot v_{res_{in}}\right] - \left[G^{N-1} d_1 + G^{N-2} d_2 + \cdots + G^0 d_N\right] \cdot V_{ref} \qquad \text{Eq. 7}$$

We can also predict the output code of the ADC by rearranging Eq. 7 into the following form

$$\frac{v_{res_{in}}}{V_{ref}} = \left[\frac{1}{G} d_1 + \frac{1}{G^{N-1}} d_2 + \cdots + \frac{1}{G^N} d_N\right] - \left[\frac{1}{G^N} \frac{v_{res_{out}}(N)}{V_{ref}}\right] \qquad \text{Eq. 8}$$

We can define the second (subtracting) term of the equation as the quantization error, and the first term as the output code x:

$$x = \left(\frac{1}{G}\right) d_1 + \left(\frac{1}{G}\right)^2 d_2 + \cdots + \left(\frac{1}{G}\right)^N d_N$$

which is exactly in the desired binary form

Algorithmic (Cyclic) ADC

Discussion:

Advantage: Conversion needs N cycles

Problem: Maintaining a constant gain of 2 may be challenging. Therefore, when G < 2, the residue plot would look like Figure b. At the same time, this adds a level of complexity to the calibration of the converter

A plot relating the residue amplifier's input and output is created for G=2and G<2 as shown below:

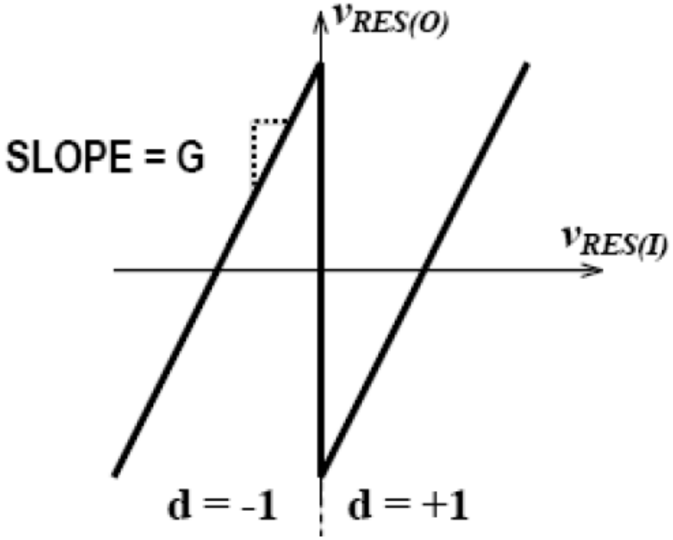


Figure a. -Residue Plot at G=2

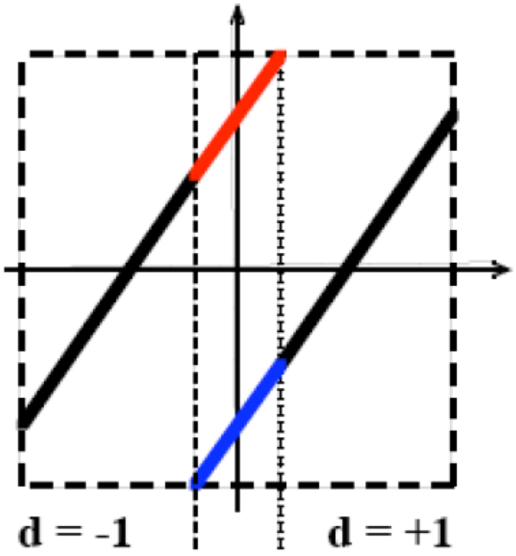
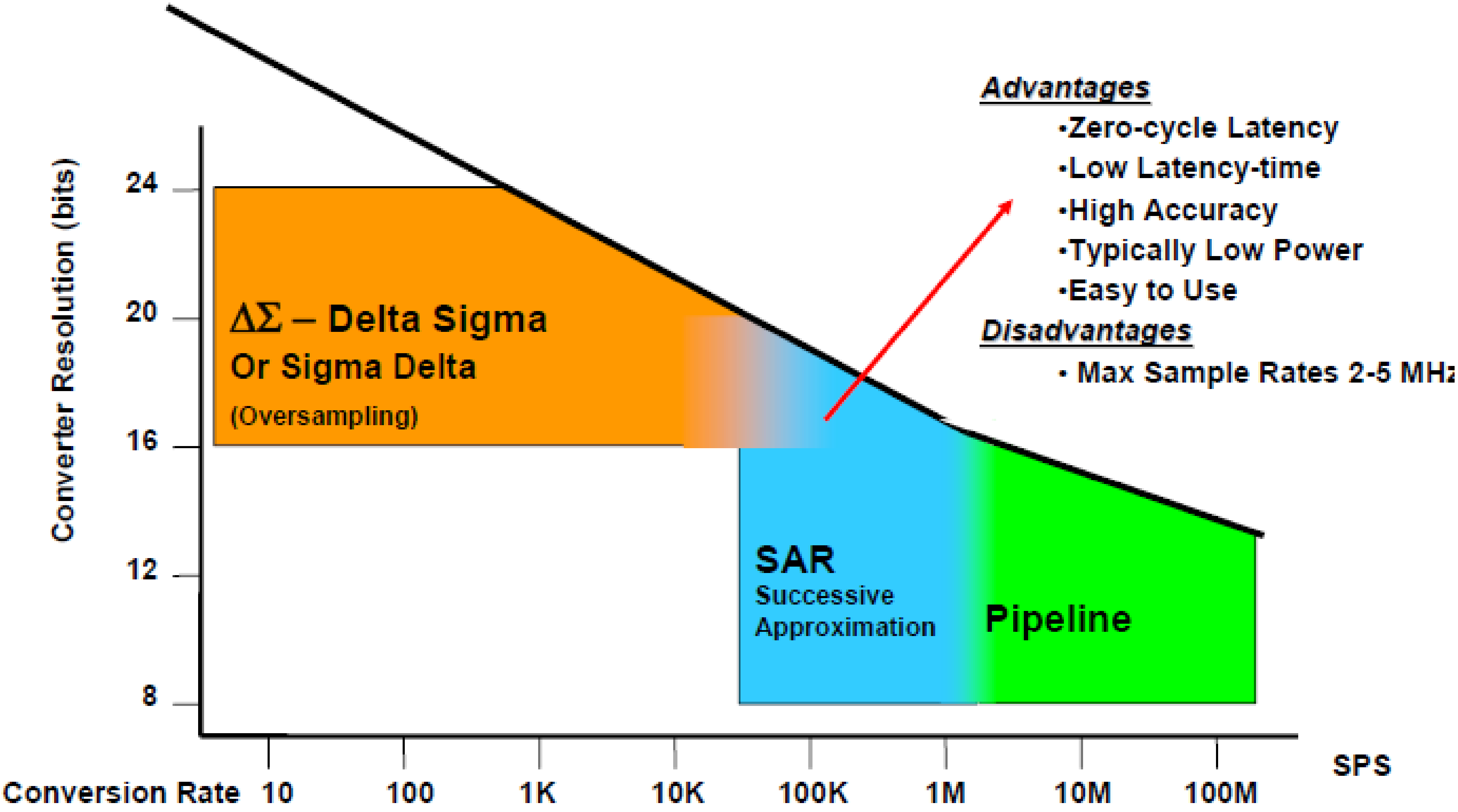


Figure b. - Residue Plot with G < 2

ADC Technologies - SAR



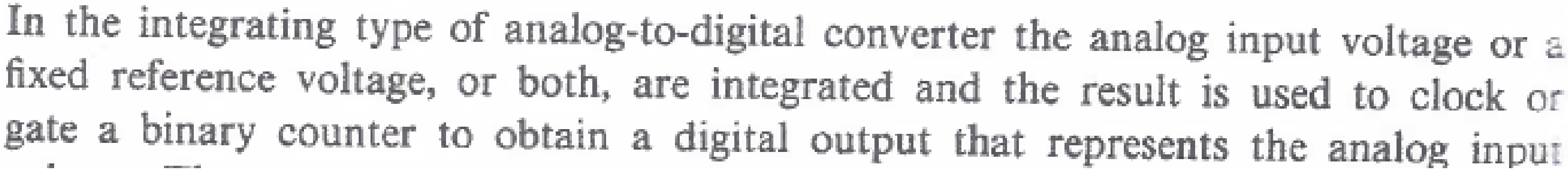
Advantages

- •Zero-cycle Latency
- •Low Latency-time
- •High Accuracy
- •Typically Low Power
- •Easy to Use

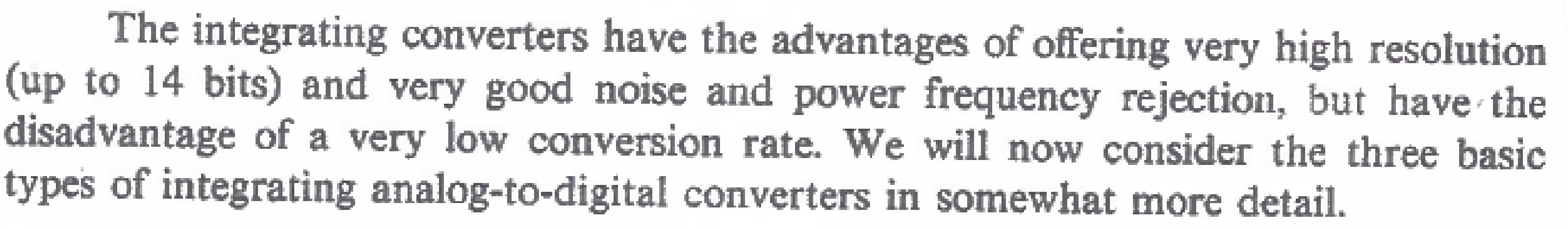
Disadvantages

- • Max Sample Rates 2-5 MHz

Integrating A/D Converters

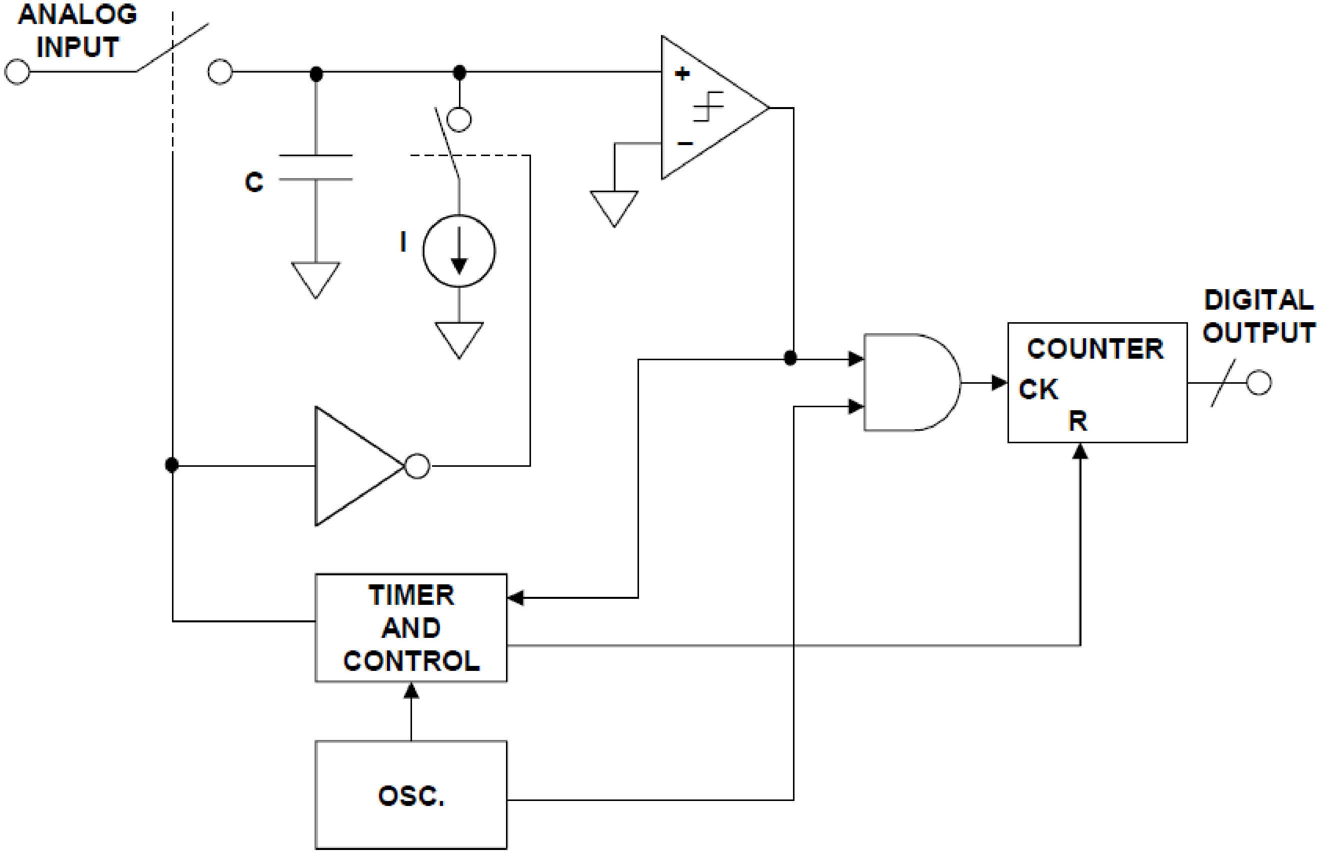


In the integrating type of analog-to-digital converter the analog input voltage or a fixed reference voltage, or both, are integrated and the result is used to clock or gate a binary counter to obtain a digital output that represents the analog input



The integrating converters have the advantages of offering very high resolution (up to 14 bits) and very good noise and power frequency rejection, but have the disadvantage of a very low conversion rate. We will now consider the three basic types of integrating analog-to-digital converters in somewhat more detail.

Charge Run-Down ADC



The charge run-down ADC architecture) shown in Figure first samples the analog input and stores the voltage on a fixed capacitor.

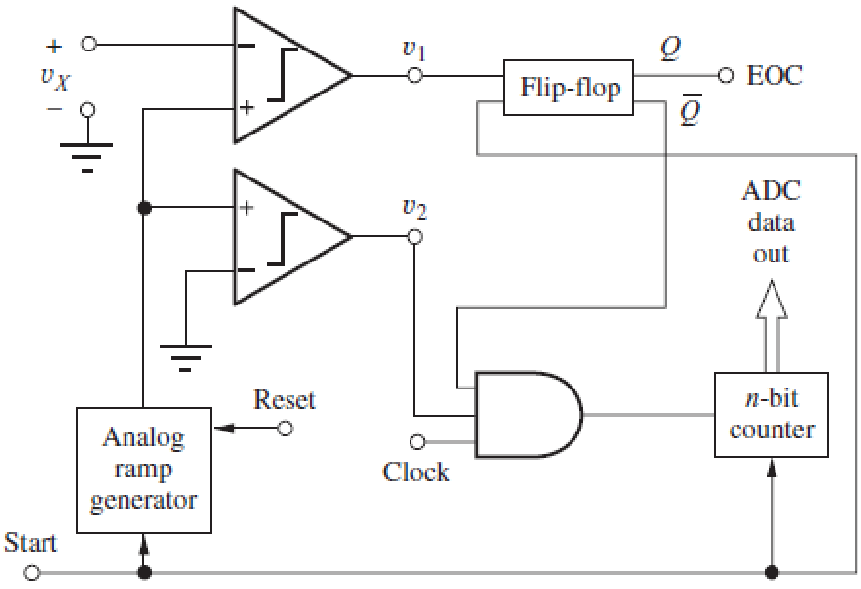
The capacitor is then discharged with a constant current source, and the time required for complete discharge is measured using a counter.

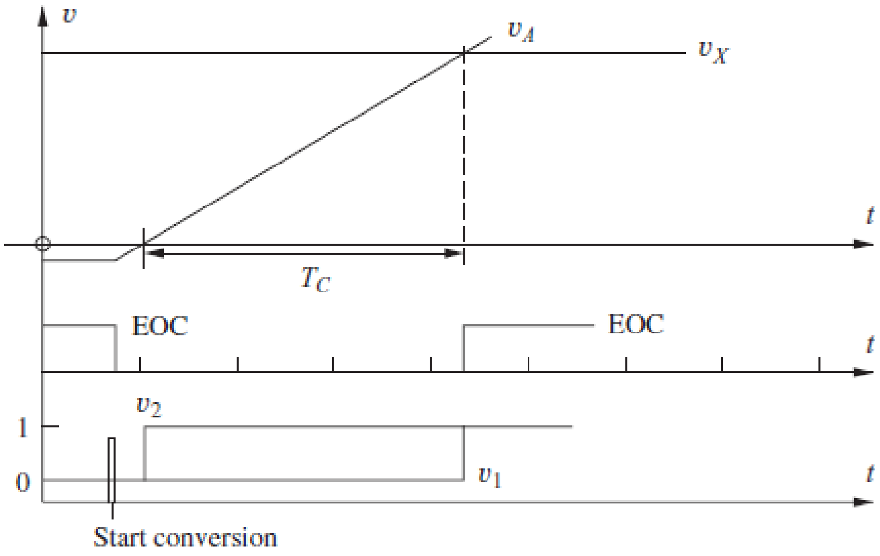
Notice that in this approach, the overall accuracy is dependent on the quality and magnitude of the capacitor, the magnitude of the current source, as well as the accuracy of the timebase.

Single-Ramp (Single-Slope) AD

Basic Concept
The discrete output of the D/A converter in the counting ADC can be replaced by a continuously changing analog reference signal. The reference voltage varies linearly with a well-defined slope from slightly below zero to above *V, and the converter is called a single-ramp,*or single-slope, ADC.
The length of time required for the reference signal to become equal to the unknown voltage is proportional to the unknown input.

Single-Ramp (Single-Slope) AD





Converter operation begins with a start conversion signal, which resets the binary counter and starts the ramp generator at a slightly negative voltage. As the ramp crosses through zero, the output of comparator 2 goes high and allows clock pulses to accumulate in the counter.

The number in the counter increases until the ramp output voltage exceeds the unknown *υχ* At this time, the output of comparator 1 goes high and prevents further clock pulses from reaching the counter.

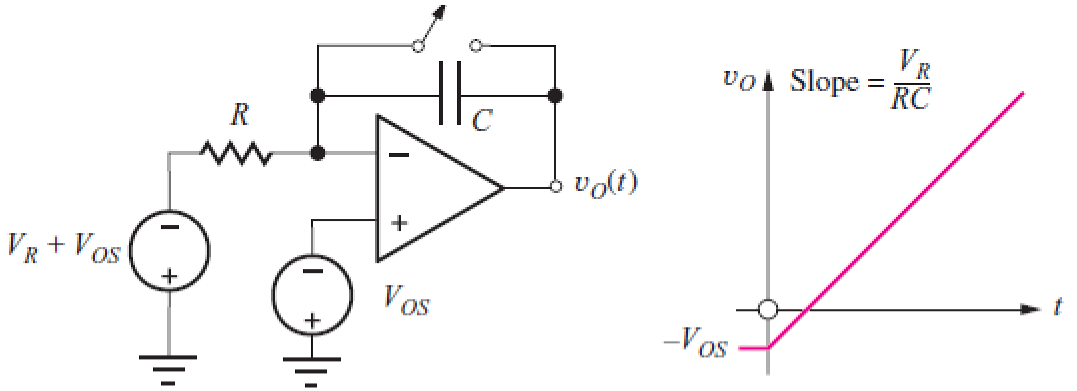
The number *N in the counter at the end of the conversion is directly proportional to the* input voltage because $\upsilon\chi = KNt_C$, where *K is the slope of the ramp in volts/second and tc is the clock period*

Single-Ramp (Single-Slope) AD

The conversion time *TT of the single-ramp converter is clearly variable and proportional to the* unknown voltage *υχ . Maximum conversion time occurs for υχ = VFS, with*

$$T_T \leq 2^n T_C$$

The counter output (N) represents the value of *υχ when* «end-of-conversion signal occurs. The ramp voltage is usually generated by an integrator connected to a constant reference voltage



When the reset switch is opened, the output increases with a constant slope given by *V_R/RC:*

$$v_O(t) = -V_{OS} + \frac{1}{RC}\int_o^t V_R\,dt$$

And thus υχ = KNt$_C$ = (V$_R$/RC) Nt$_C$

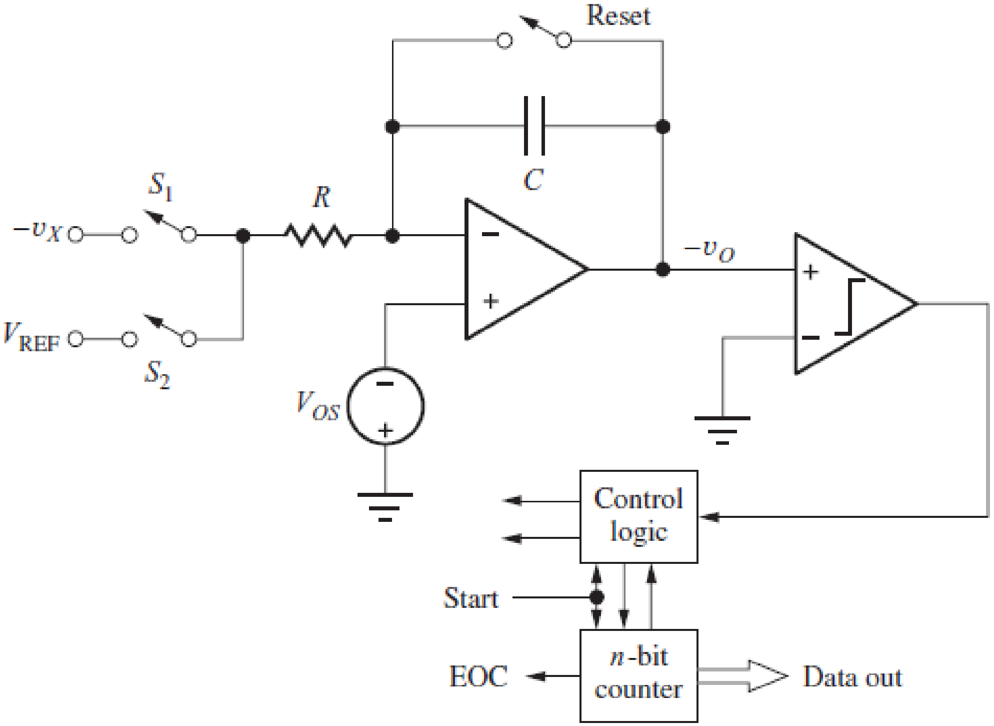
The dependence of the ramp's slope on the *RC product is one of the major limitations of the* single-ramp A/D converter. The slope depends on the absolute values of *R and C, which are difficult* to maintain constant in the presence of temperature variations and over long periods of time.

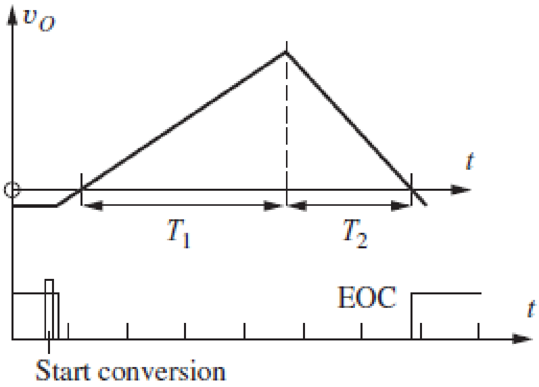
Dual-Ramp (Dual-Slope) ADC

The **dual-ramp, or dual-slope, ADC solves the problems associated with the single-ramp converter** and is commonly found in high-precision data acquisition and instrumentation systems.

The conversion cycle consists of two separate integration intervals.
-First, the unknown voltage *υχ is integrated for a known period of time T1.*
-The value of this integral is then compared to that of a known reference voltage *VREF, which is integrated for a variable length* of time *T2.*

Dual-Ramp (Dual-Slope) ADC



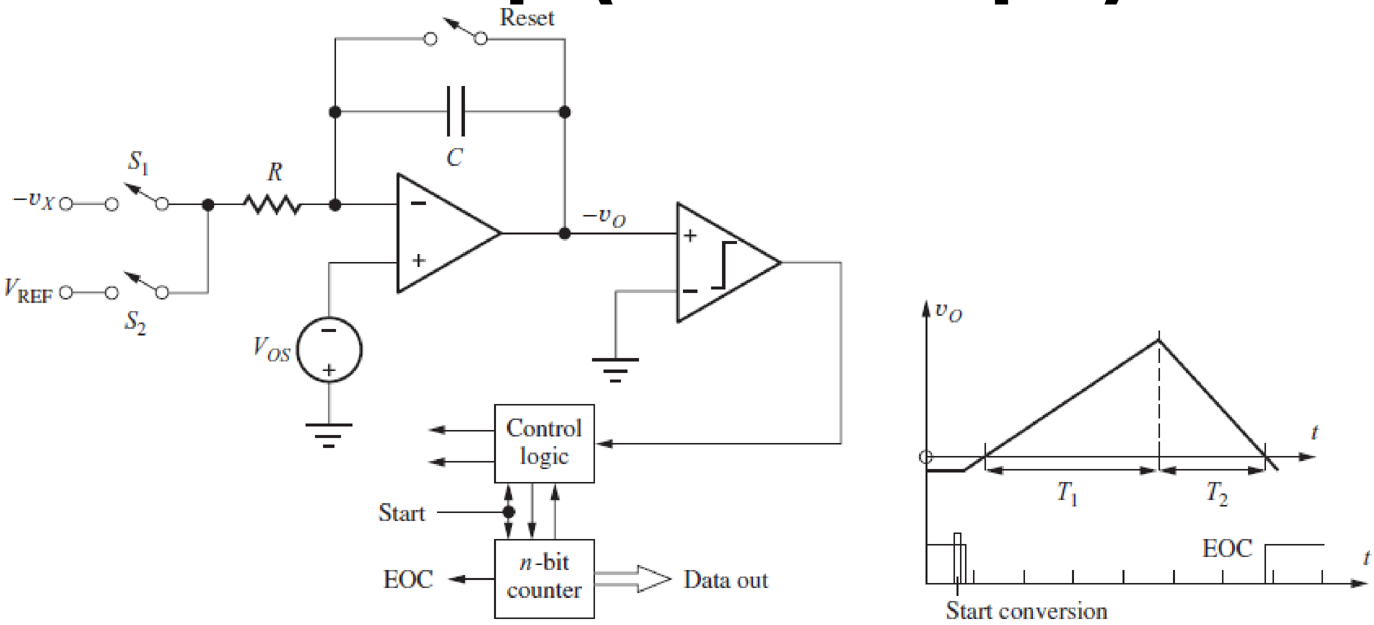


At the start of conversion the counter is reset, and the integrator is reset to a slightly negative voltage. The unknown input *υχ is connected to the integrator input through switch S1. Voltage υχ is integrated for a fixed period of time T1 = 2exp(n)T*$_C$*, which begins when the integrator* output crosses through zero. At the end of time *T1, the counter overflows, causing S1 to be opened* and *VREF to be connected to the integrator input through S2.*

The integrator output then decreases until it crosses back through zero, and the comparator changes state, indicating the end of the conversion.

The counter continues to accumulate pulses during the down ramp, and the final number in the counter represents the quantized value of the unknown voltage *vX* .

Dual-Ramp (Dual-Slope) ADC



Circuit operation forces the integrals over the two time periods to be equal:

$$\frac{1}{RC}\int_0^{T_1} v_X(t)\,dt = \frac{1}{RC}\int_{T_1}^{T_1+T_2} V_{\text{REF}}\,dt$$

T1 is set equal to 2 exp(n)T_C *because the unknown voltage υχ was integrated over the amount of time* needed for the *n-bit counter to overflow. Time period T2 is equal to N*T_C*, where N is the number* accumulated in the counter during the second phase of operation. It holds that:

$$\frac{1}{RC}\int_0^{T_1} v_X(t)\,dt = \frac{\langle v_X\rangle}{RC}T_1 \qquad \frac{1}{RC}\int_{T_1}^{T_1+T_2} V_{\text{REF}}(t)\,dt = \frac{V_{\text{REF}}}{RC}T_2$$

and finally:

$$\frac{\langle v_X\rangle}{V_{\text{REF}}} = \frac{T_2}{T_1} = \frac{N}{2^n}$$

And thus , indeed N is proportional to analog voltage υχ

Dual-Ramp (Dual-Slope) ADC

The absolute values of *R and C no longer enter directly into the relation between vX and VRef, and the* long-term stability problem associated with the single-ramp converter is overcome.

Furthermore, the digital output word represents the average value of *υχ during the first integration phase. Thus, υχ can change during the conversion cycle of this converter without destroying the validity of the* quantized output value.

The conversion time T_T *requires 2expn clock periods for the first integration period, and N clock* periods for the second integration period. Thus the conversion time is variable and

$$T_T = (2^n + N)T_C \leq 2^{n+1}T_C$$ since the maximum value of *N=2expn*

Important error might be introduced though by the offset voltage of the op-amp In this case the error in final measurement (N of the counter) could be expressed as :

$$Kerr = 2^N\left(\frac{Vin}{Vref} - \frac{Vin - Vos}{Vref + Vos}\right) = 2^N\left(\frac{Vos\cdot(Vin + Vref)}{Vref\cdot(Vref + Vos)}\right)$$

The max error is derived when Vin =Vref giving

$$Kerr = 2^N\left(\frac{2\cdot Vos}{(Vref + Vos)}\right) \approx 2^N\left(\frac{2\cdot Vos}{Vref}\right)$$

In order that the measurement (value N of counter) not be affected it should hold that Kerr<1 , giving

$$Vos \leq \frac{Vref}{2^{N+1}} = \frac{1}{2}V_{LSB}$$

Dual-Ramp (Dual-Slope) ADC

Conclusion: Advantages- Disadvantages

The dual ramp is a widely used converter. Although much slower than the successive approximation converter, the dual-ramp converter offers excellent differential and integral linearity.

By combining its integrating properties with careful design, one can obtain accurate conversion at resolutions exceeding 20 bits, but at relatively low conversion rates.

In a number of recent converters and instruments, the basic dual-ramp converter has been modified to include extra integration phases for automatic offset voltage elimination

High Speed ADCs

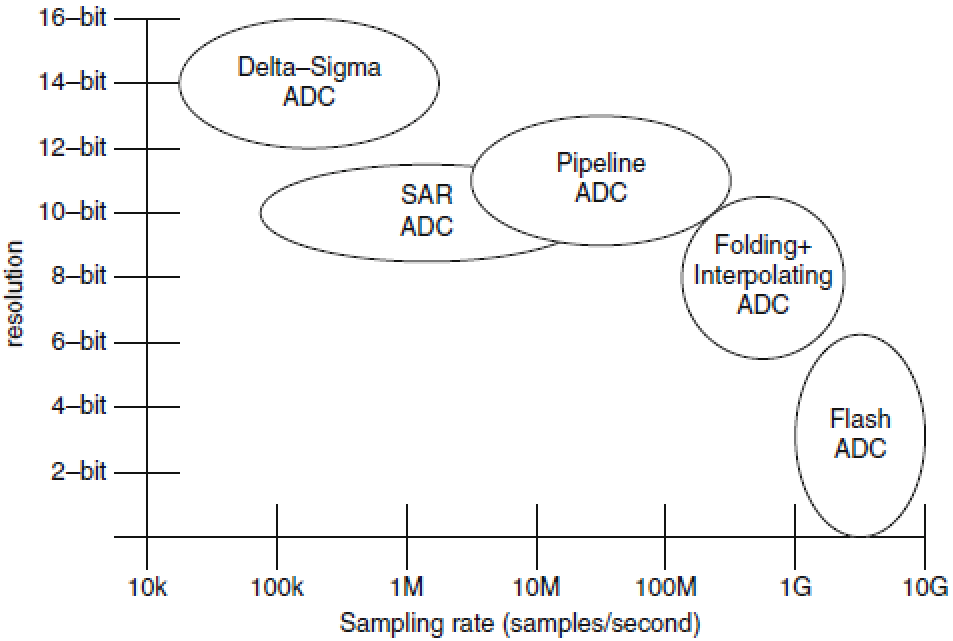
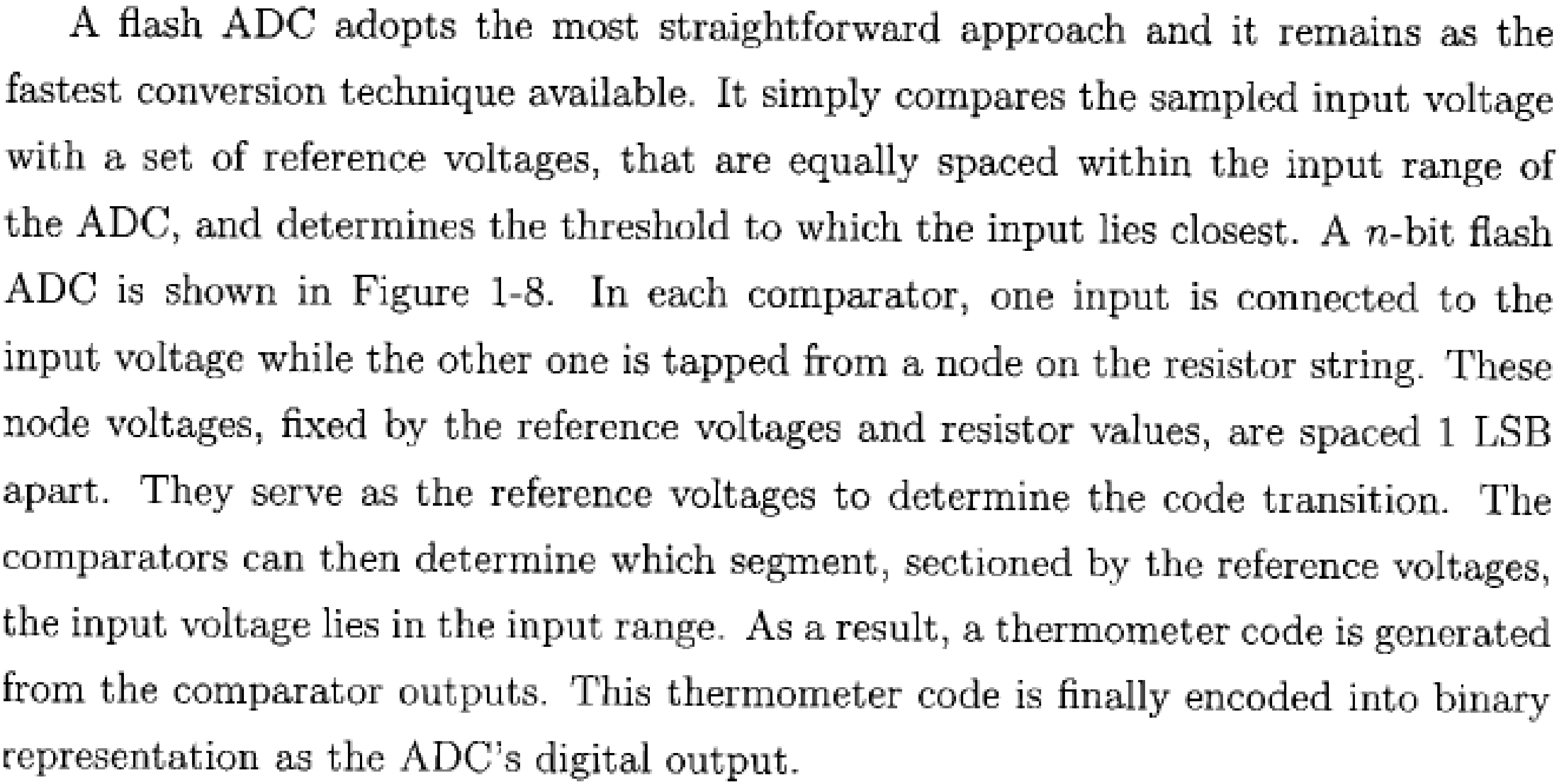


Table 2.1 Comparison of ADC architectures

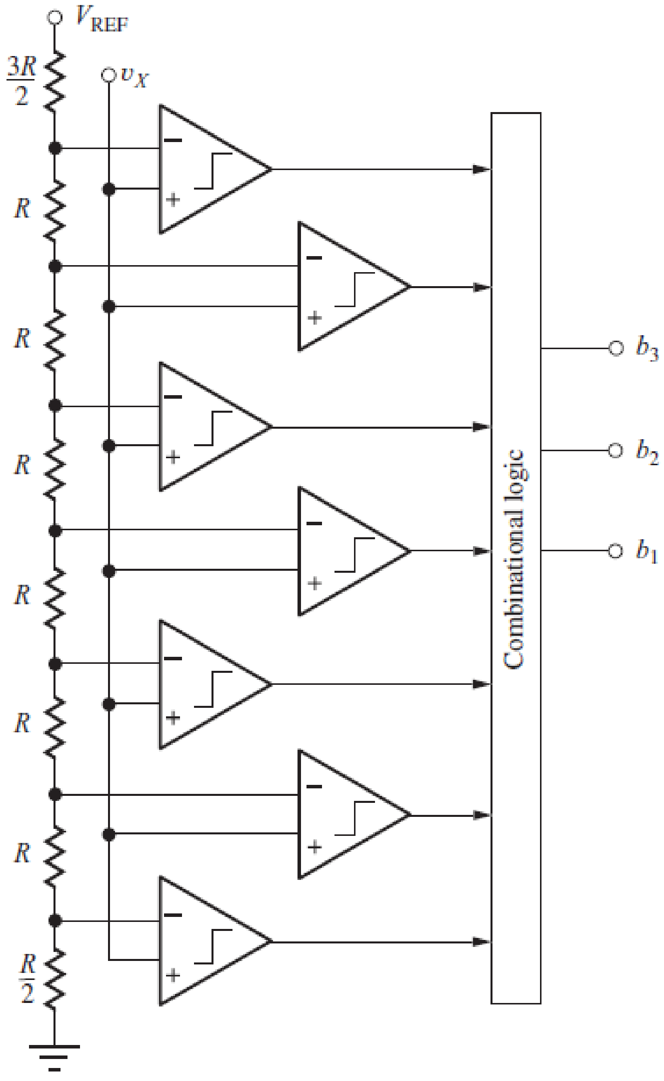
Architecture	Latency	Speed	Accuracy	Area
Flash	Low	High	Low	High
SAR	Low	Low-medium	Medium-high	Low
Folding + interpolating	Low	Medium-high	Medium	High
Delta-sigma	High	Low	High	Medium
Pipeline	High	Medium-high	Medium-high	Medium

Flash ADCs-Basic Mechanism



A flash ADC adopts the most straightforward approach and it remains as the fastest conversion technique available. It simply compares the sampled input voltage with a set of reference voltages, that are equally spaced within the input range of the ADC, and determines the threshold to which the input lies closest. A n-bit flash ADC is shown in Figure 1-8. In each comparator, one input is connected to the input voltage while the other one is tapped from a node on the resistor string. These node voltages, fixed by the reference voltages and resistor values, are spaced 1 LSB apart. They serve as the reference voltages to determine the code transition. The comparators can then determine which segment, sectioned by the reference voltages, the input voltage lies in the input range. As a result, a thermometer code is generated from the comparator outputs. This thermometer code is finally encoded into binary representation as the ADC's digital output.

Flash ADC



the unknown input *vX is simultaneously compared to seven different reference voltages.* The logic network encodes the comparator outputs directly into three binary bits representing the quantized value of the input voltage.

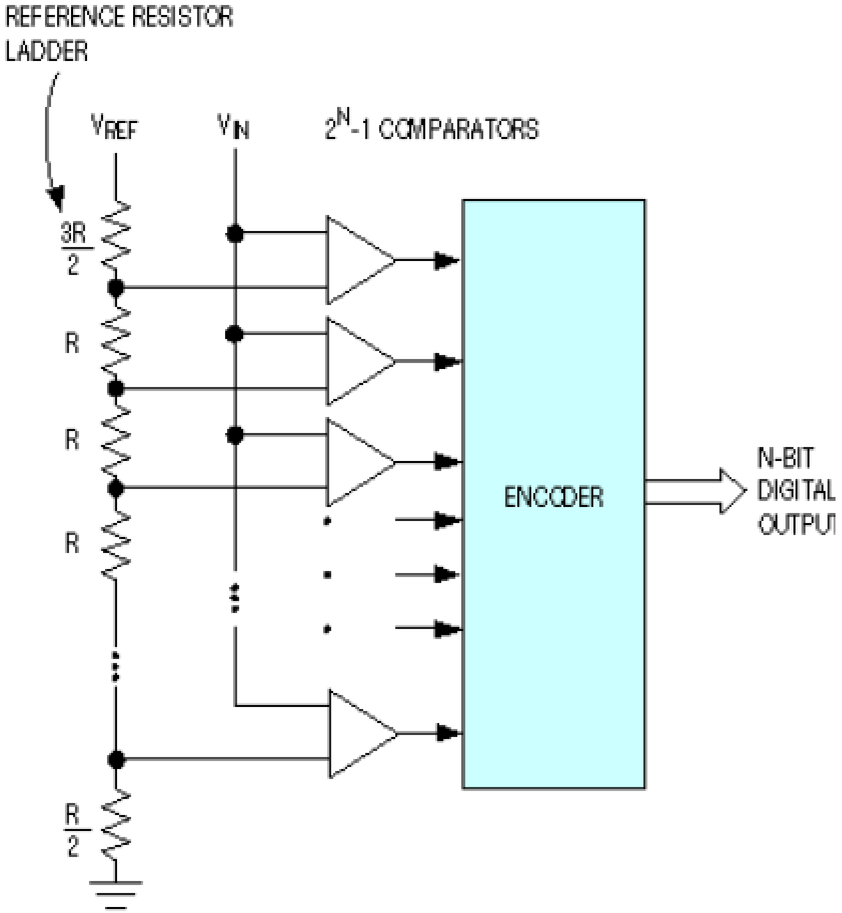
The speed of this converter is very fast, limited only by the time limited delays of the comparators and logic network. Also, the output continuously reflects the input signal

delayed by the comparator and logic network.

The parallel A/D converter is used when maximum speed is needed and is usually found in converters with resolutions of 10 bits or less because 2exp(*n* − 1) *comparators and reference voltages* are needed for an *n-bit converter. Thus the cost of implementing such a converter grows* rapidly with resolution. However, converters with 6-, 8-, and 10-bit resolutions have been realized

in monolithic IC technology. These converters achieve effective conversion rates as high as 10exp8–10exp9 conversions/second

Flash ADCs –Basic Structure



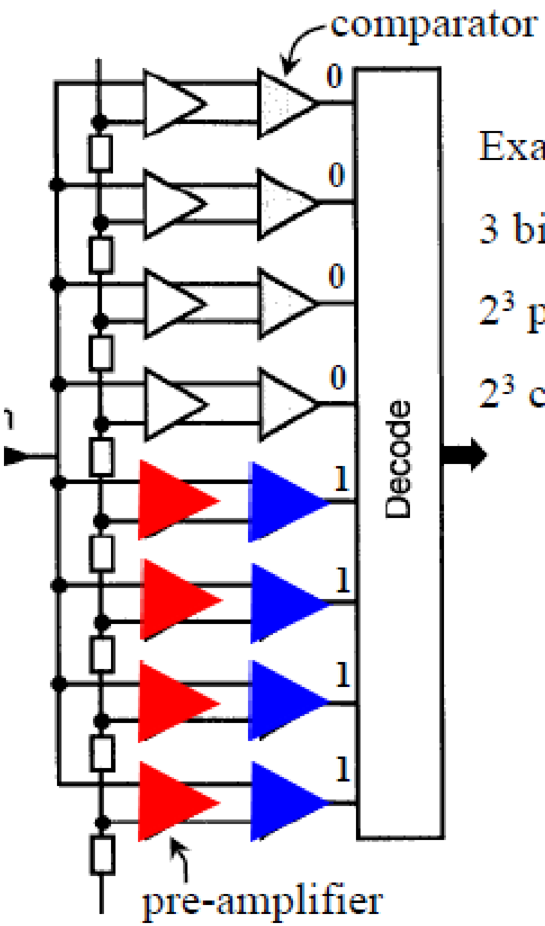
Flash ADC Structure-Characteristics

Best up to 8 bits:

+ Speed

+ Simplicity

- Exponential complexity
- Big input capacitance
- Bubbles in thermo code
- Power
- Difference in signal delay to each comparator



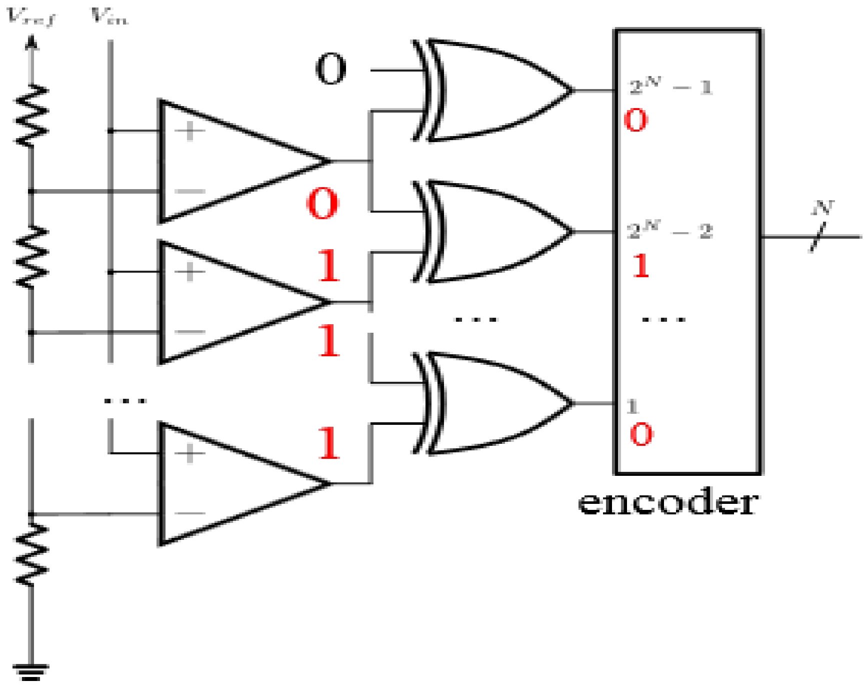
Example:

3 bit Flash ADC

2^3 pre-amplifiers

2^3 comparators

Flash ADCs-The priority encoder

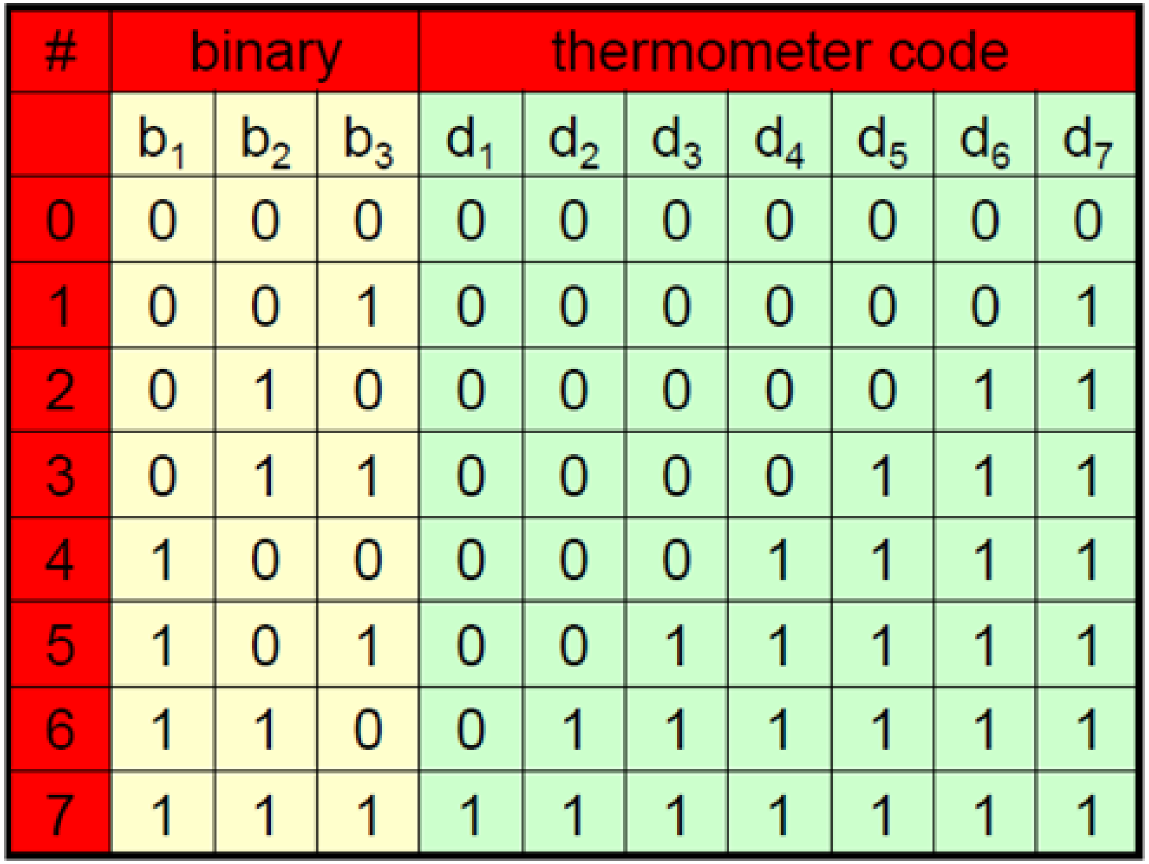


The Priority Encoder

The Priority Encoder has to find the position of the last comparator with high output, starting from the bottom. That means that it should find the position where neighboring comparators have different outputs (all below have output high and all above have output low).

That can be simply done by XORing the outputs of neighboring comparators and feeding their outputs to a digital encoder. Only one XOR has its output active and the encoder will translate that position into a binary representation. If there are 2expN comparators, the encoder outputs a N-bit number.

Thermometer Code Converters (method to force monotonicity)



#	binary			thermometer code						
	b_1	b_2	b_3	d_1	d_2	d_3	d_4	d_5	d_6	d_7
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	0	0	0	0	0	1	1
3	0	1	1	0	0	0	0	1	1	1
4	1	0	0	0	0	0	1	1	1	1
5	1	0	1	0	0	1	1	1	1	1
6	1	1	0	0	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1

Flash ADC Pros and Cons

Strength: Flash converters are the fastest types available (sampling rates to GHz), covering the analog signal to digital word instantly

Weakness: Their resolution is constrained by the available die size and by excessive input capacitance and power consumption caused by the large number of comparators used. More specifically:

•Their repetitive structure demands precise matching between the parallel comparator sections, because any mismatch can cause static error such as a magnified input offset voltage (or current).

•Flash ADCs are also prone to sporadic and erratic outputs known as "sparkle codes" with two major sources:

-Metastability in the 2exp (N-1) comparators

-Thermometer-code bubbles

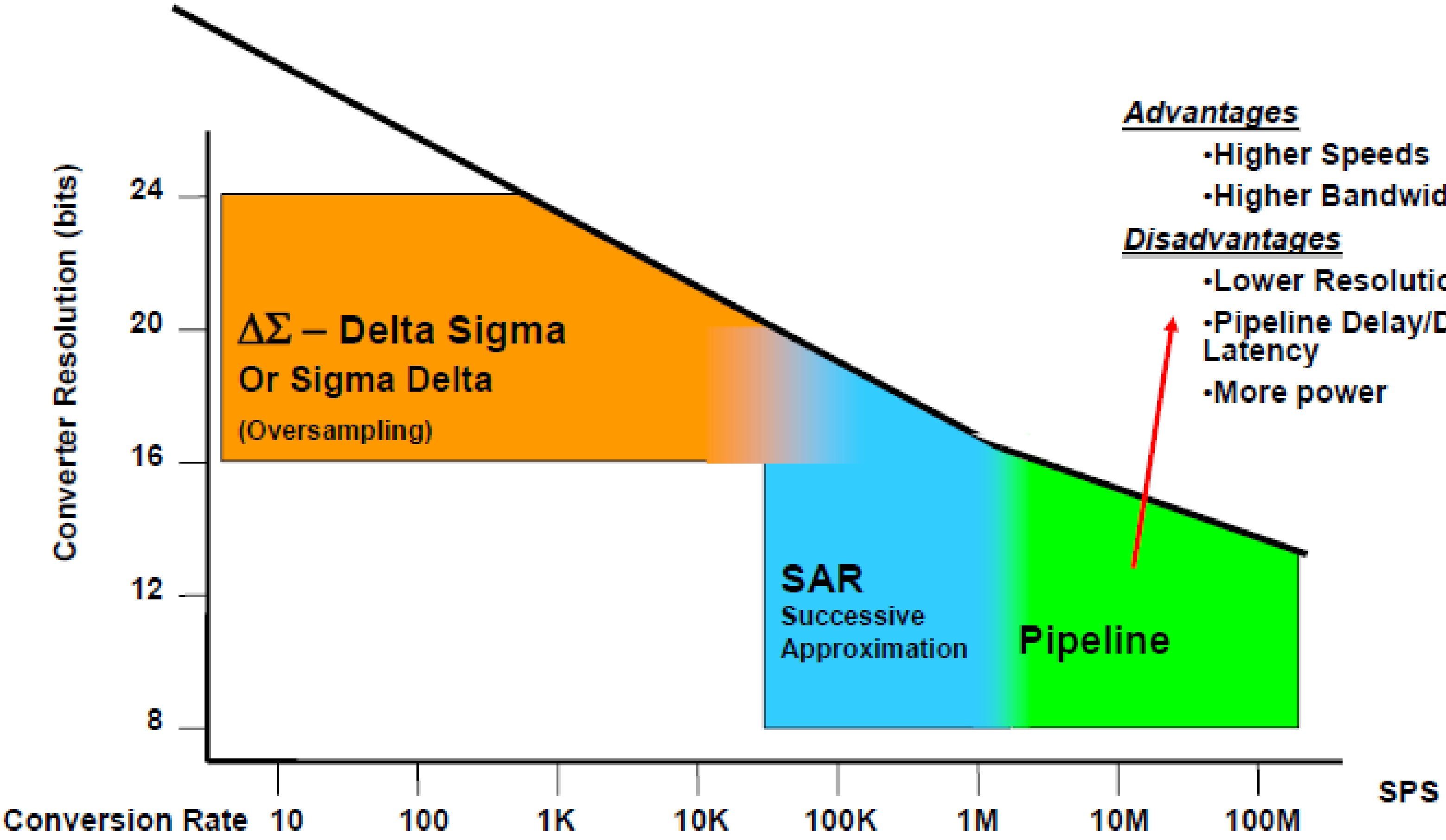
•Mismatched comparator delays can turn a logical 1 into 0 (or vice versa), causing the appearance of "bubbles" in an otherwise normal thermometer code. Because the ADC's encoder unit cannot detect this error, it generates an out-of-sequence code that also appears as an output "spark."

•Another concern with flash ADCs is its die size, which is nearly seven times larger for an 8-bit flash converter than for the equivalent pipelined ADC.

•In further contrast to pipeline designs, the flash converter's input capacitance can be six times higher and its power dissipation twice as high.

Subrange-Pipelined ADCs

ADC Technologies - pipeline



Advantages

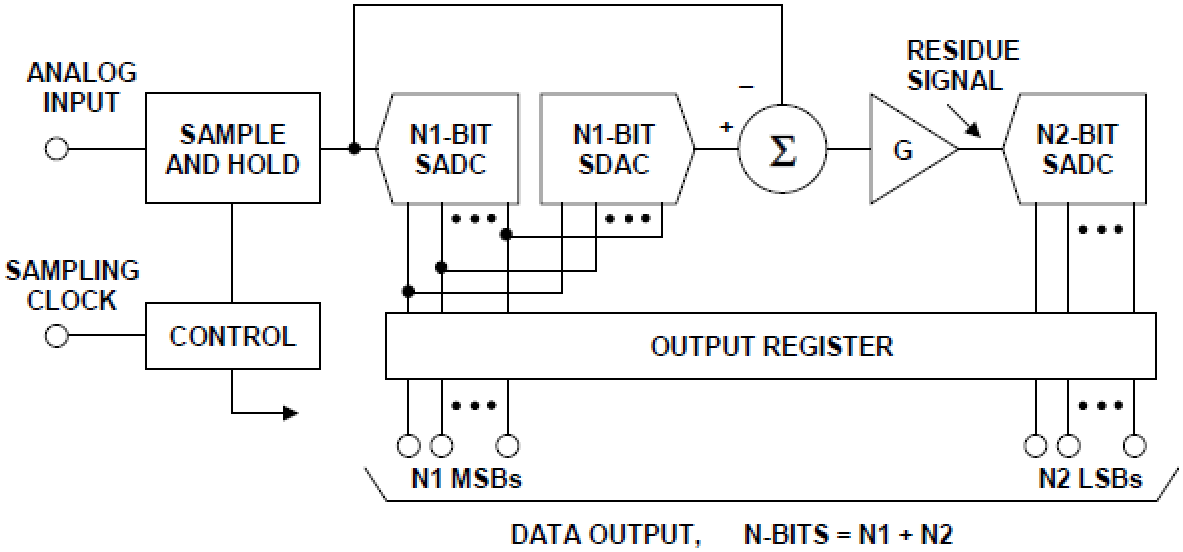
- Higher Speeds
- Higher Bandwidth

Disadvantages

- Lower Resolution
- Pipeline Delay/Data Latency
- More power

Two- Step ADC

N-bit Two-Stage Subranging ADC



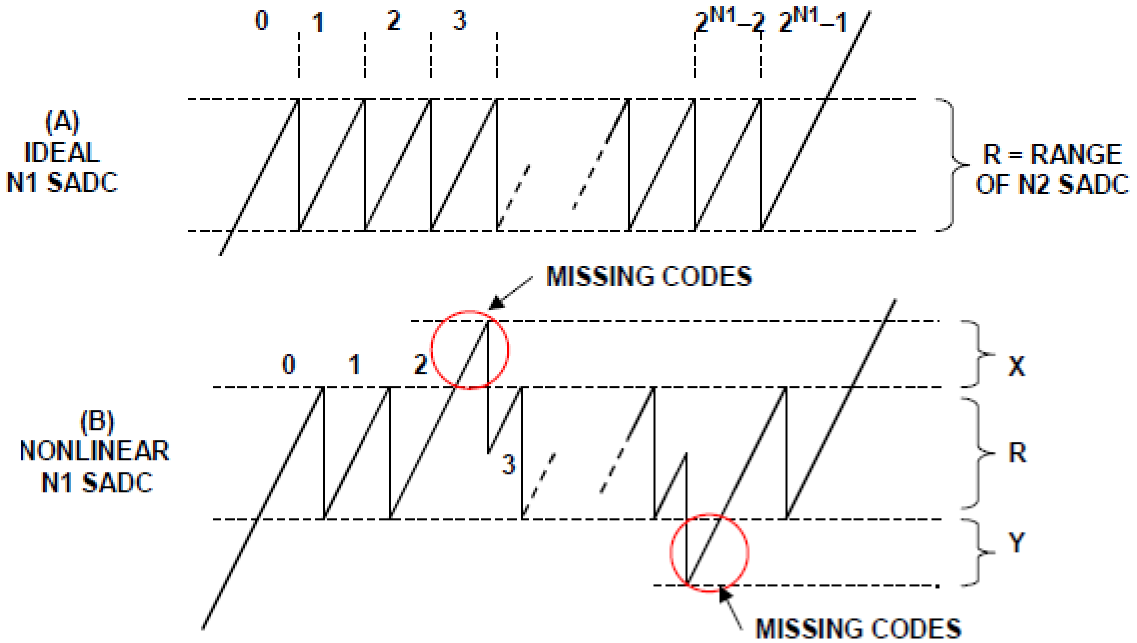
See: R. Staffin and R. Lohman, "Signal Amplitude Quantizer,"
U.S. Patent 2,869,079, Filed December 19, 1956, Issued January 13, 1959

Basic Mechanism: The conversion process begins placing the sample-and-hold in the hold mode followed by a coarse N1-bit sub-ADC (SADC) conversion of the MSBs. +
The digital outputs of the MSB converter drive an N1-bit sub-DAC (SDAC) which generates a coarsely quantized version of the analog input signal. The N1-bit SDAC output is subtracted from the held analog signal, amplified, and applied to the N2-bit LSB SADC.
The amplifier provides gain, G = 2exp (N1), sufficient to make the "residue" signal exactly fill the input range of the N2 SADC. The output data from the N1 SADC and the N2 SADC are latched into the output registers yielding the N-bit digital output code, where N = N1 + N2.
No of comparators =2exp N1+2exp N2< Flash ADCs (expN)

- ADC can be best analyzed by examining the residue waveform at the input to the second-stage ADC
- In order for there to be no missing codes, the residue waveform must exactly fill the input range of the second-stage ADC
- Both the N1 ADC and the N1 DAC must be better than N1 + N2 bits accurate—in the example shown, N1 = 3, N2 = 3, and N1 + N2 = 6
- The situation shown in Figure 2B in next slide will result in missing codes when the residue waveform goes outside the range of the N2 ADC, "R", and falls within the "X" or "Y" regions—caused by a nonlinear N1 ADC or interstage gain and/or offset mismatch.

Subrange- Pipelined ADCs

Figures (A), (B): *Residue Waveforms at Input of N2 Sub-ADC*

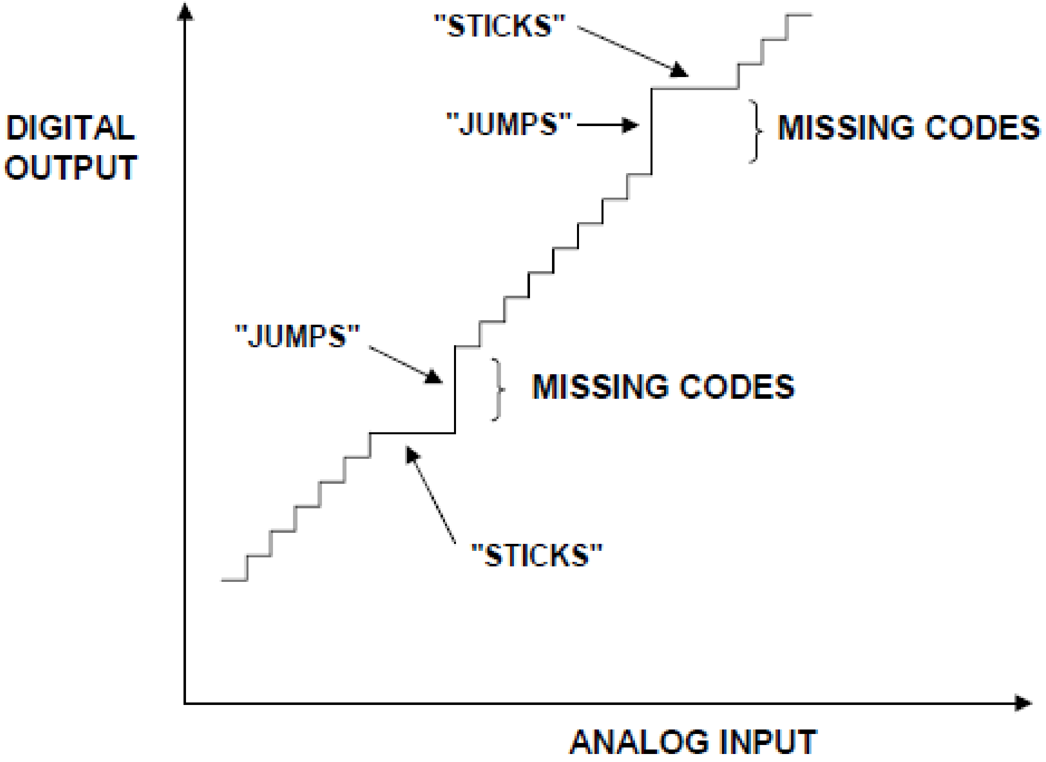


In order for this simple subranging architecture to work satisfactorily, both the N1 SADC and especially the SDAC must be better than N-bits accurate. The residue signal offset and gain must be adjusted such that it precisely fills the range of the N2 SADC as shown in Figure above.

If the residue signal drifts by more than 1 LSB (referenced to the N2 SADC), then there will be missing codes as shown in next slide where the residue signal enters the out-of-range regions labeled "X" and "Y". Any nonlinearity or drift in the N1 SADC will also cause missing codes if it exceeds 1 LSB referenced to N-bits.

Subrange- Pipelined ADCs

Missing Codes Due to MSB SADC Nonlinearity or Interstage *Misalignment*



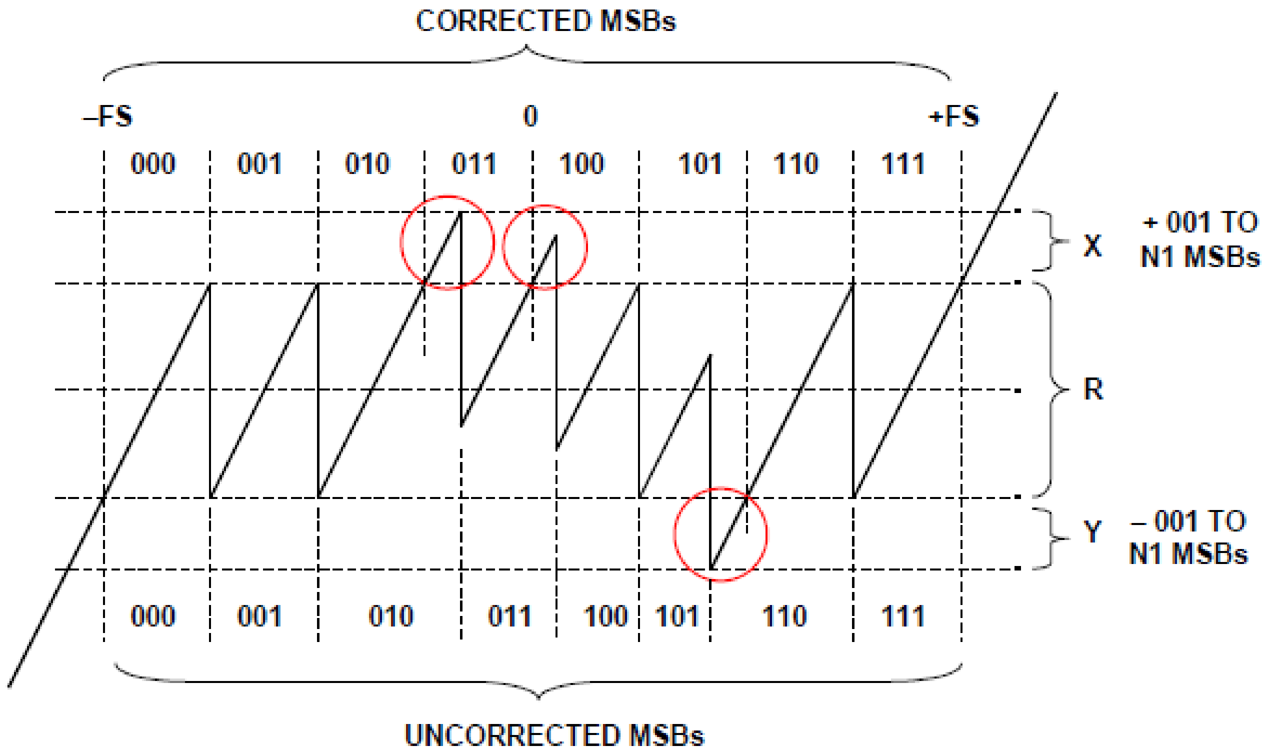
When the interstage alignment is not correct, missing codes will appear in the overall ADC transfer function as shown in Figure above. If the residue signal goes into positive overrange (the "X" region), the output first "sticks" on a code and then "jumps" over a region leaving missing codes. The reverse occurs if the residue signal is negative overrange.

In practice, an 8-bit subranging ADC with N1 = 4 bits and N2 = 4 bits represents a realistic limit to this architecture in order to maintain no missing codes over a reasonable operating temperature range

- In order to reliably achieve higher than 8-bit resolution, a technique generally referred to as *digitally corrected subranging, digital error correction, overlap bits, redundant bits,* etc. is utilized.
- The fundamental concept is illustrated using the residue waveform shown in Figure next slide.

Subrange- Pipelined ADCsx

Solution: Error Correction Using Added Quantization Levels (for N1 = 3)



For example in the two-stage 6-bit subranging ADC, extra quantization levels in the positive and the negative overrange region X, Y are added (equivalently an extra bit is added) to the second-stage ADC which allows the digitization of the regions shown as “X” and “Y” in Figure above.

The extra range in the second-stage ADC allows the residue waveform to deviate from its ideal value-provided it does not exceed the range of the second-stage ADC.

- The residue waveform is shown for the specific case where N1 = 3 bits.
- In a standard ADC, the residue waveform must exactly fill the input range of the N2 ADC—it must stay within the region designated R.
- The missing code problem is solved by adding extra quantization levels
  - in the positive overrange region X and
  - the negative overrange region Y
- These additional levels require additional comparators in the basic N2 flash ADC.
- Modern digitally corrected subranging ADCs generally obtain the additional quantization levels by using an internal ADC with higher resolution for the N2 ADC.
- For instance, if one additional bit is added to the N2 ADC, its range is doubled—then the residue waveform can go outside either end of the range by ½ LSB referenced to the N1 ADC

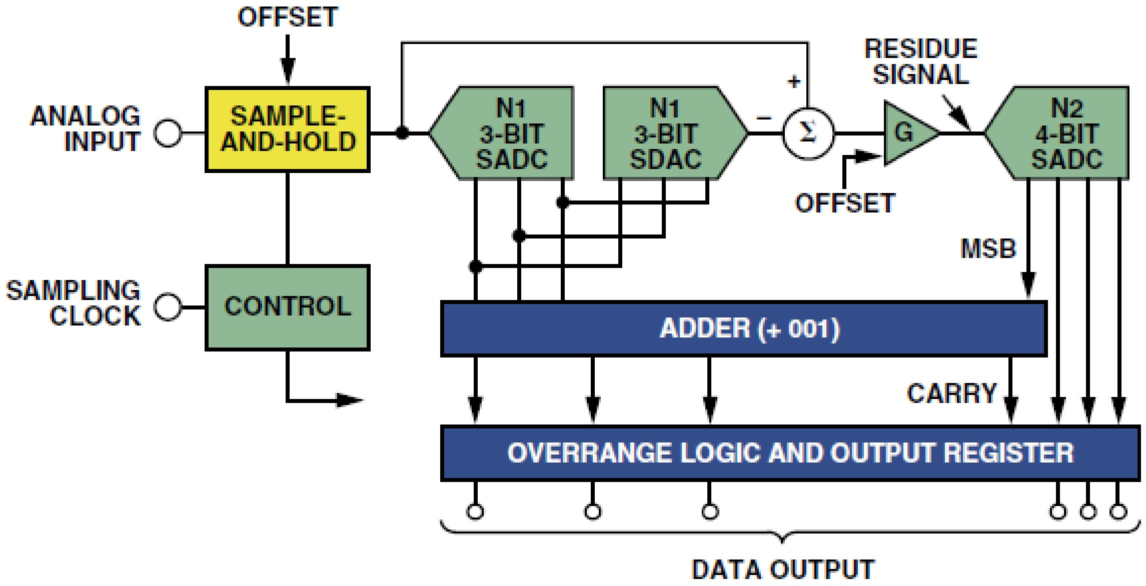
- There is no theoretical reason why more bits can't be added to the second stage, thereby allowing more errors in the first stage, but practical design considerations and tradeoffs come into play here.

- In practice, rather than adding or subtracting 001 to the MSBs, an offset can be added to the residue signal so that the MSBs are either passed through to the output unmodified, or with 001 added to them. This simplifies the logic.

Subrange- Pipelined ADCs

Two-step ADC with extra bit in second ADC stage for error correction (missing codes avoidance)

6-bit subranging error-corrected ADC example , N1 = 3, N2 = 4.



A basic 6-bit subranging ADC with error correction is shown in Figure with the second-stage resolution increased to 4 bits, rather than the original 3 bits.

Additional logic, required to modify the results of the N1 SADC when the residue waveform falls in the “X” or “Y” overrange regions, is implemented with a simple adder in conjunction with a dc offset voltage added to the residue waveform. In this arrangement, the MSB of the second-stage SADC controls whether the MSBs are incremented by 001 or passed through unmodified. The carry output of the adder is used in conjunction with some simple overrange logic to prevent output bits from returning to all-zeros state when the input signal goes outside the positive range of the ADC.

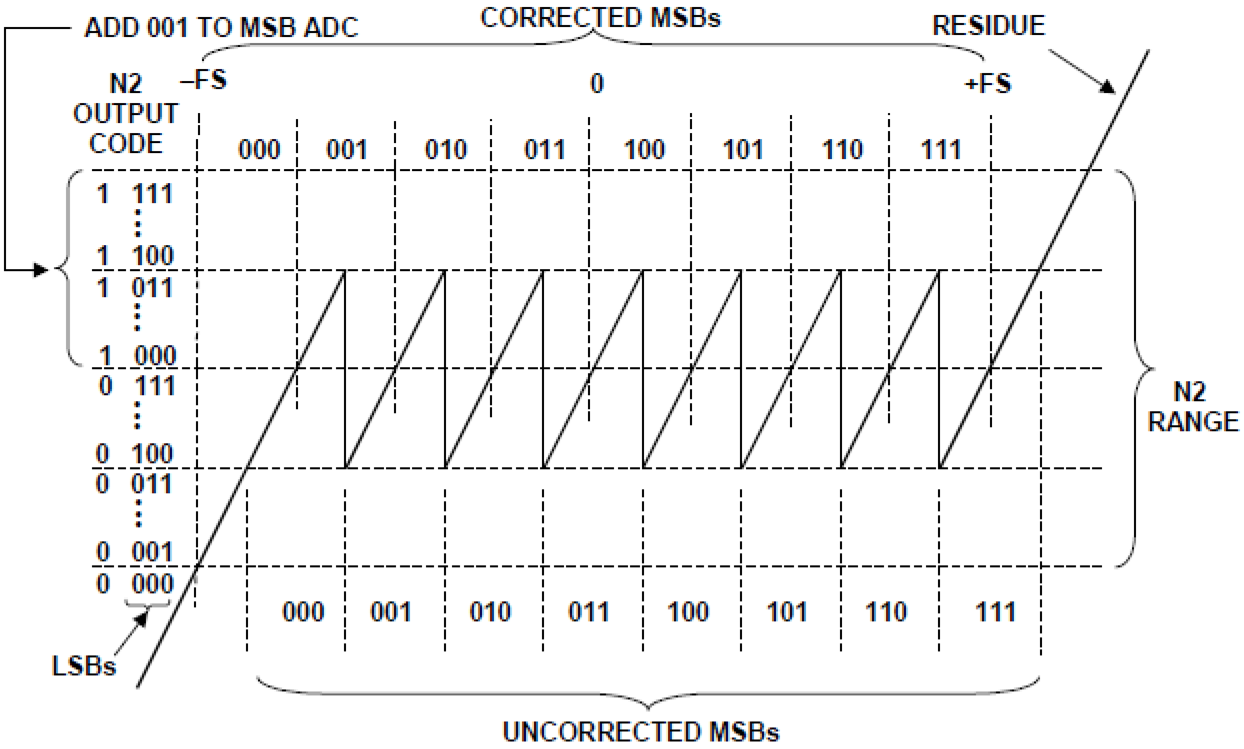
More than one correction bit can be used in the 2nd-stage ADC, a trade-off—part of the converter design process

- After passing through an input sample-and-hold, the signal is digitized by the 3-bit ADC, reconstructed by a 3-bit DAC, subtracted from the held analog signal and then amplified and applied to the second 4-bit ADC
- The gain of the amplifier, G, is chosen so that the residue waveform occupies ½ the input range of the 4-bit ADC.
- The 3 LSBs of the 6-bit output data word go directly from the second ADC to the output register

- The MSB of the 4-bit ADC controls whether or not the adder adds 001 to the 3 MSBs.
- The carry output of the adder is used in conjunction with some simple overrange logic to prevent the output bits from returning to the all-zeros state when the input signal goes outside the positive range of the ADC.
- Figure on the next slide shows the ideal residue waveform assuming perfect linearity in the first ADC and perfect alignment between the two stages.

Subrange- Pipelined ADCs

Two-step ADC with extra bit in second ADC stage for error correction (missing codes avoidance)



6-Bit Error Corrected Subranging ADC N1 = 3, N2 = 4, Ideal MSB SADC

Notice that the gain of the amplifier, G, is chosen so that the residue waveform occupies ½ the input range of the 4-bit SADC

- Notice that the residue waveform occupies exactly ½ the range of the N2 ADC

- Following the residue waveform from left-to-right
  - as the input first enters the overall ADC range at –FS, the N2 ADC begins to count up, starting at 0000
  - When the N2 ADC reaches the 1000 code, 001 is added to the N1 ADC output causing it to change from 000 to 001
- As the residue waveform continues to increase, the N2 ADC continues to count up until it reaches the code 1100, at which point the N1 ADC switches to the next level
- The DAC switches and causes the residue waveform to jump down to the 0100 output code
- The adder is now disabled because the MSB of the N2 ADC is zero, so the N1 ADC output remains 001. The residue waveform then continues to pass through each of the remaining regions until +FS is reached.

Subrange- Pipelined ADCs

Two-step ADC with extra bit in second ADC stage for error correction (missing codes avoidance)

6-Bit Error Corrected Subranging ADC 1 = 3 , N2 = 4, Nonlinear MSB SADC

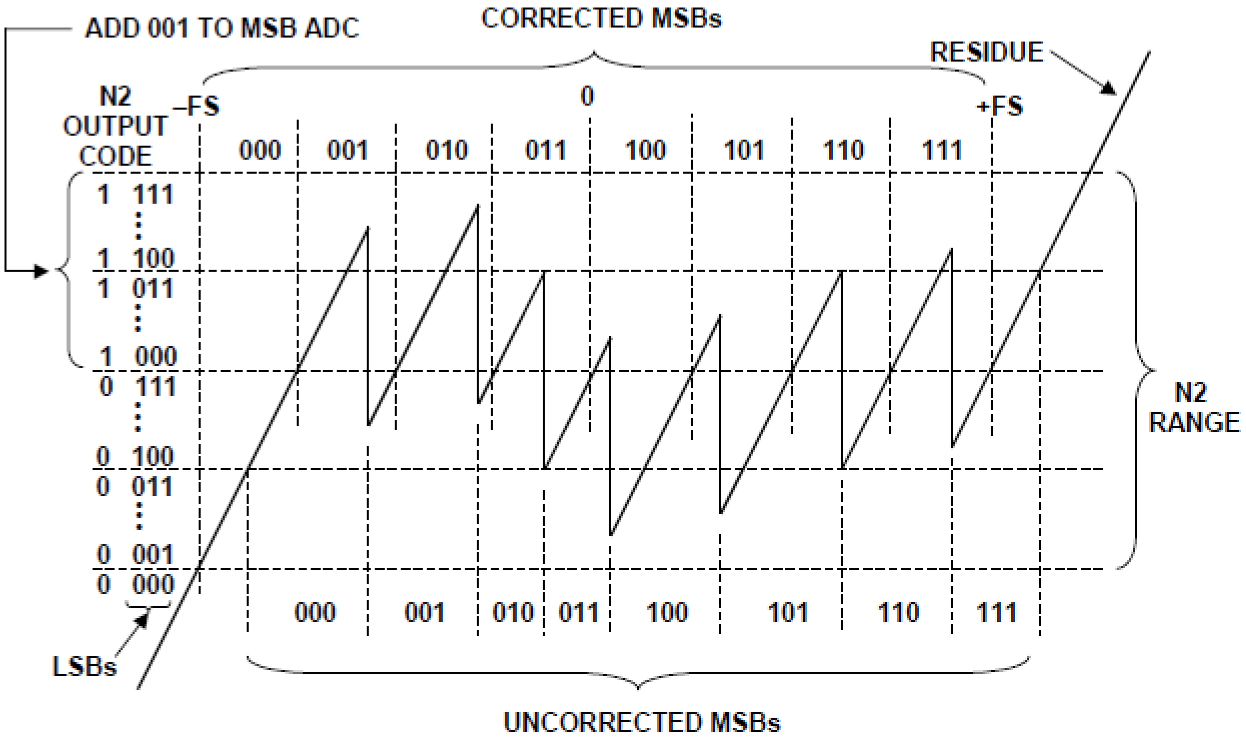


Figure shows a residue signal where there are errors in the N1 SADC. Notice that there is no effect on the overall ADC linearity provided the residue signal remains within the range of the N2 SADC.

As long as this condition is met, the error correction method described corrects for the following errors: *sample-and-hold droop error, sample-and hold settling time error, N1 SADC gain error, N1 SADC offset error, N1 SDAC offset error, N1 SADC linearity error, residue amplifier offset error – but not for gain and linearity of the N1 SDAC and amplifier*

Fully Pipelined ADCs (with identical stages)

Basic Pipelined ADC with Identical Stages

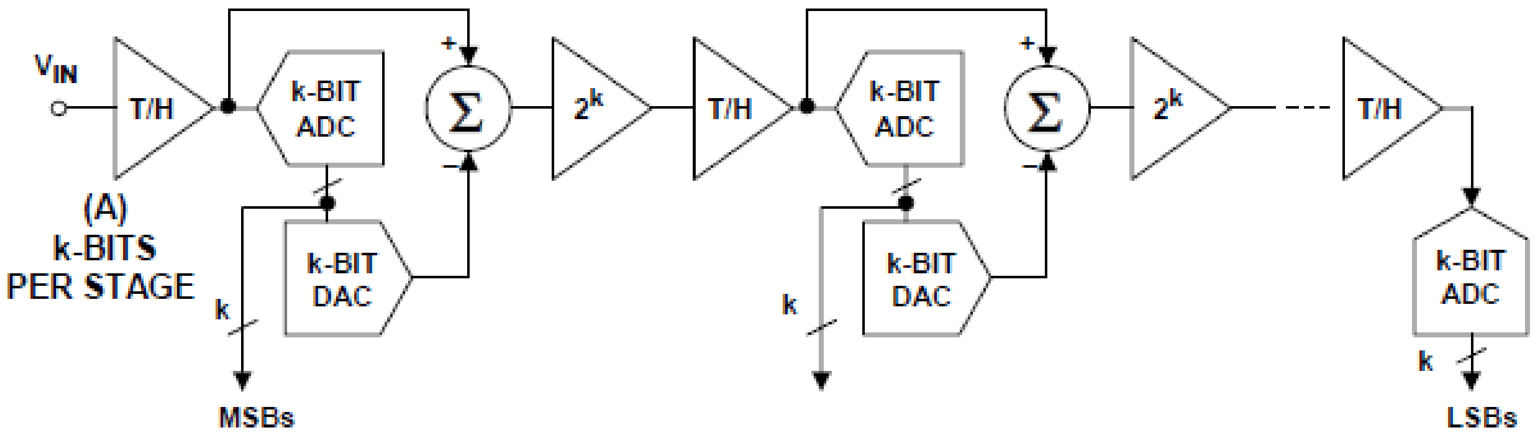
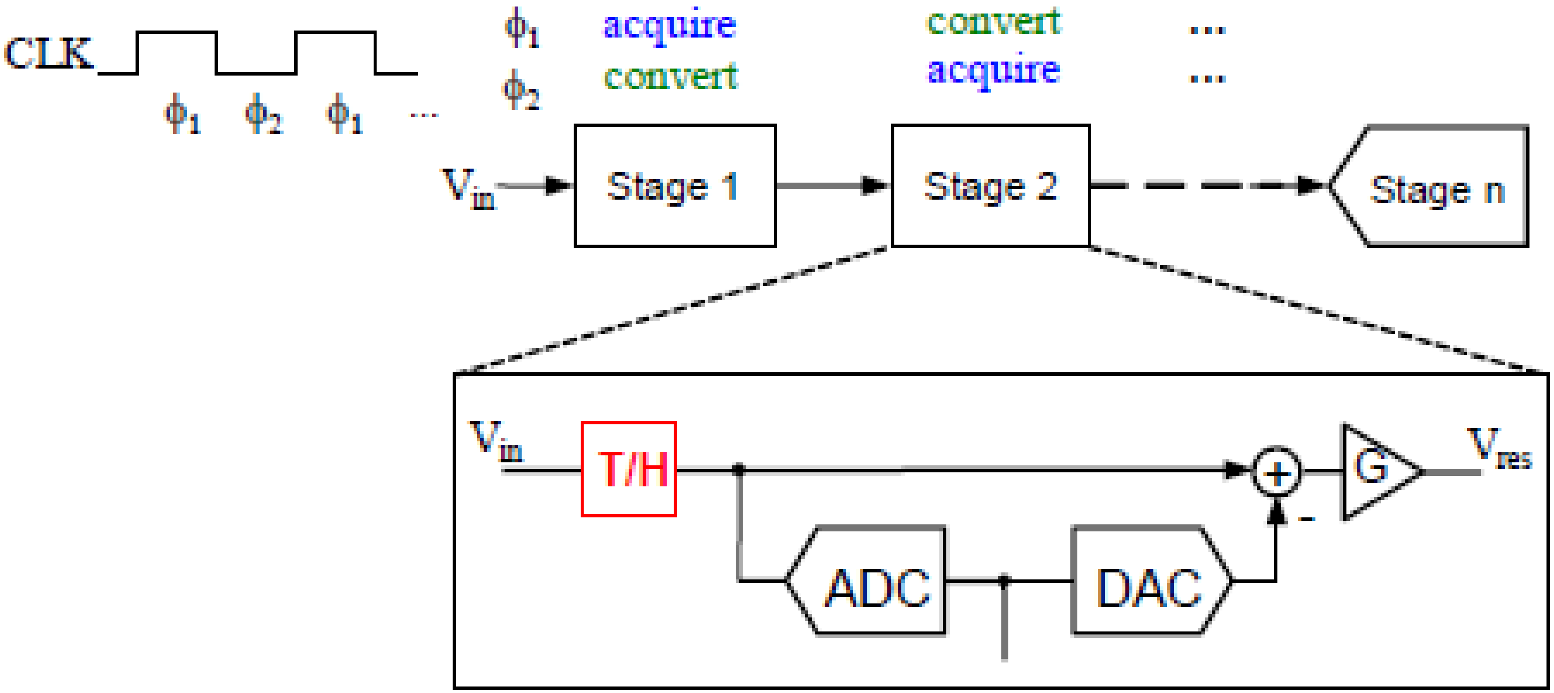


Figure above shows pipelined stages which use an interstage T/H (Track and Hold i.e Sample and Hold) and give each stage the maximum possible amount of time to process the signal at its input.

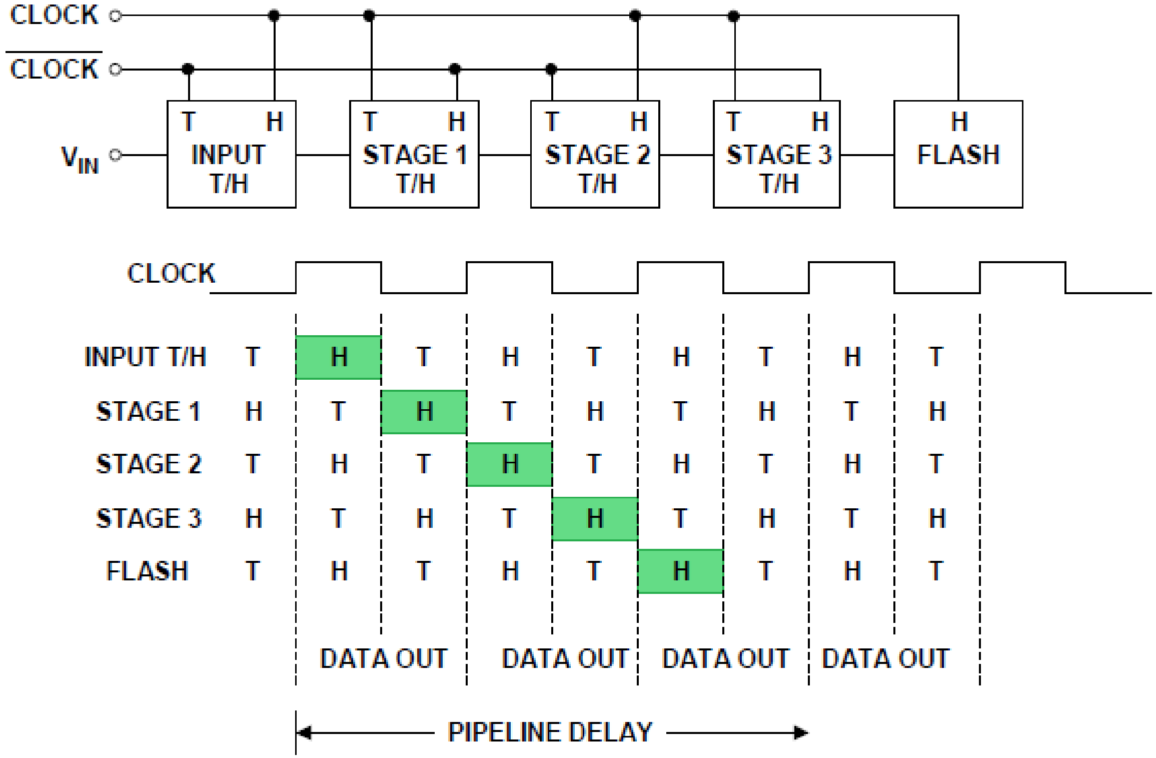
The term "pipelined" architecture refers to the ability of one stage to process data from the previous stage during any given clock cycle. At the end of each phase of a particular clock cycle, the output of a given stage is passed on to the next stage using the T/H functions and new data is shifted into the stage.
Of course this means that the digital outputs of all but the last stage in the "pipeline" must be stored in the appropriate number of shift registers so that the digital data arriving at the correction logic corresponds to the same sample

Pipelined ADC
Stage Implementation


CLK
ϕ_1 ϕ_2 ϕ_1 ...
ϕ_1 acquire convert ...
ϕ_2 convert acquire ...
V_{in}
Stage 1
Stage 2
Stage n
V_{in}
T/H
+
-
G
V_{res}
ADC
DAC

- Each stage needs T/H hold function
- Track phase: Acquire input/residue from previous stage
- Hold phase: sub-ADC decision, compute residue

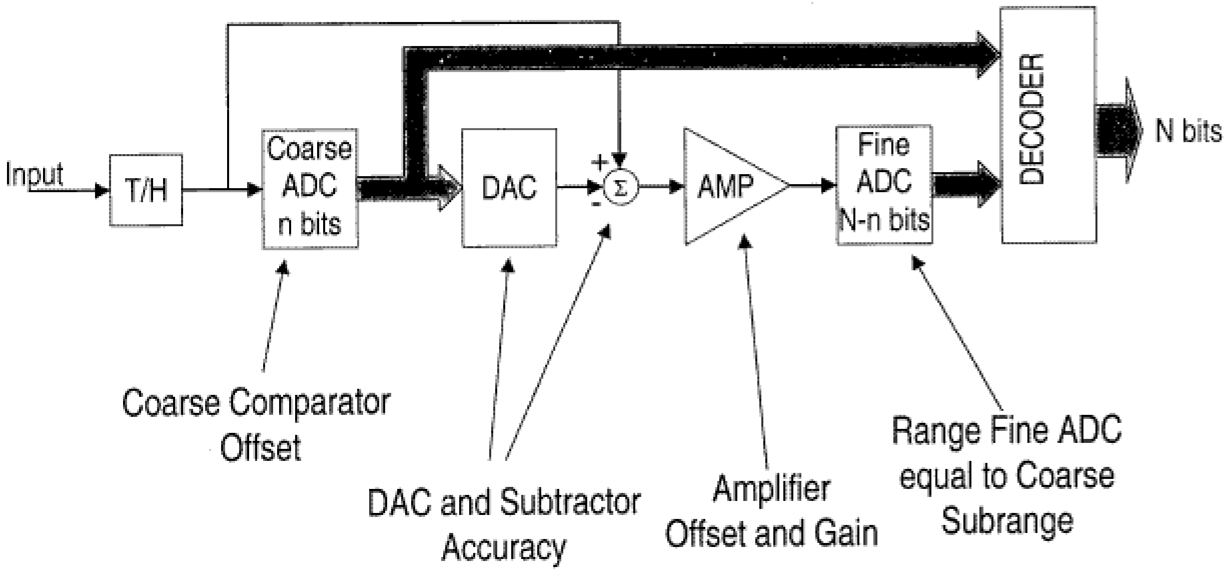
Fully Pipelined ADCs (with identical stages)



INPUT T/H	T	H	T	H	T	H	T	H	T
STAGE 1	H	T	H	T	H	T	H	T	H
STAGE 2	T	H	T	H	T	H	T	H	T
STAGE 3	H	T	H	T	H	T	H	T	H
FLASH	T	H	T	H	T	H	T	H	T

Figure shows timing of a typical pipelined subranging ADC. The phases of the clocks to the T/H amplifiers are alternated from stage to stage such that when a particular T/H in the ADC enters the hold mode it holds the sample from the preceding T/H, and the preceding T/H returns to the track mode. The held analog signal is passed along until it reaches the final stage. When operating at high sampling rates, it is critical that the differential sampling clock be kept at a 50% duty cycle for optimum performance. Duty cycles other than 50% affect all the T/H amplifiers in the chain—some will have longer than optimum track times and shorter than optimum hold times; while others suffer the reverse condition. Newer pipelined ADCs have on-chip clock conditioning circuits to control internal duty cycle allowing some variation in external clock duty cycle

Position of Accuracy Errors in ADC pipelined stages


Input
T/H
Coarse ADC n bits
DAC
+
-
Σ
AMP
Fine ADC N-n bits
DECODER
N bits
Coarse Comparator Offset
DAC and Subtractor Accuracy
Amplifier Offset and Gain
Range Fine ADC equal to Coarse Subrange

Digital Error Correction for Pipelined ADCs
Bits Combination

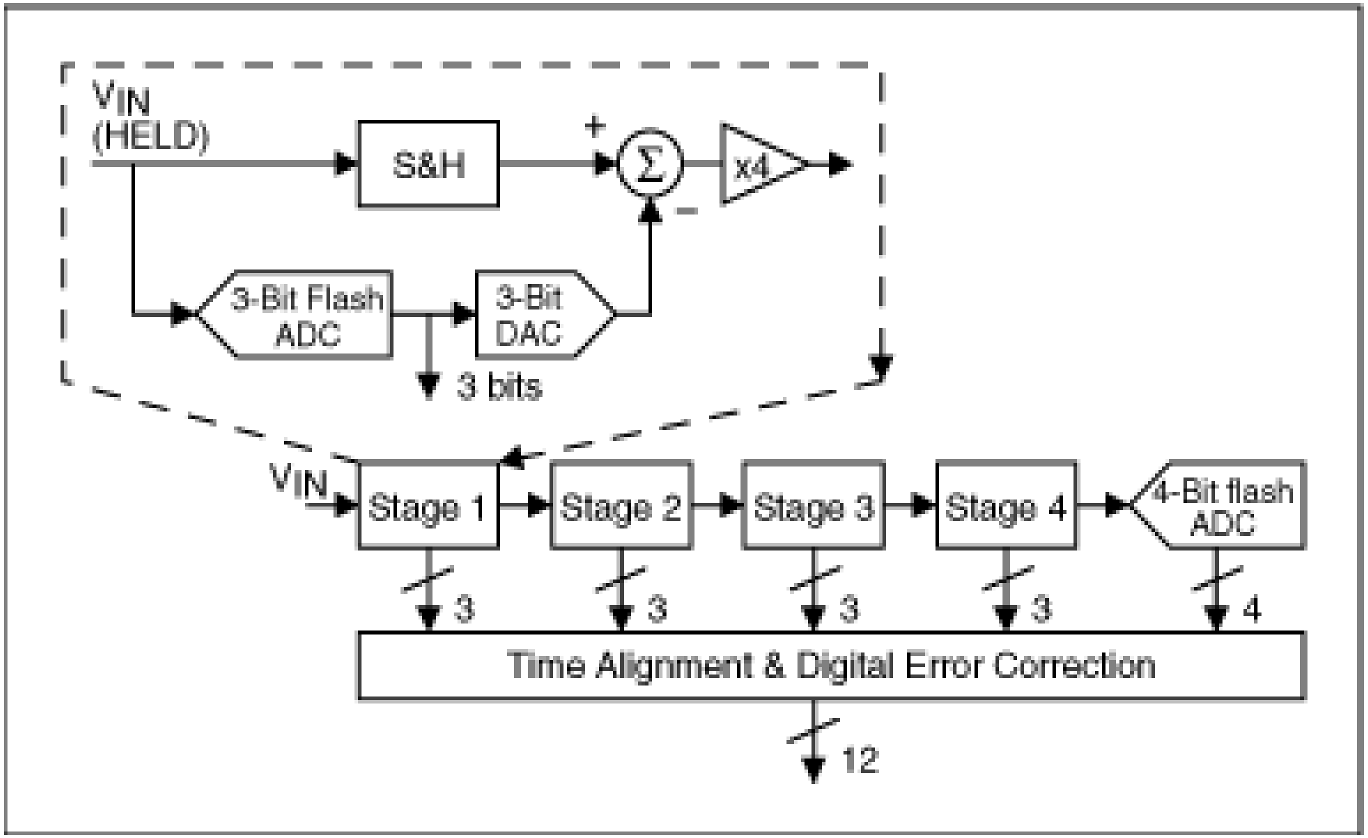


Figure 1. Pipelined ADC with four 3-bit stages (each stage resolves two bits).

Digital Error Correction for Pipelined ADCs -Bits Combination

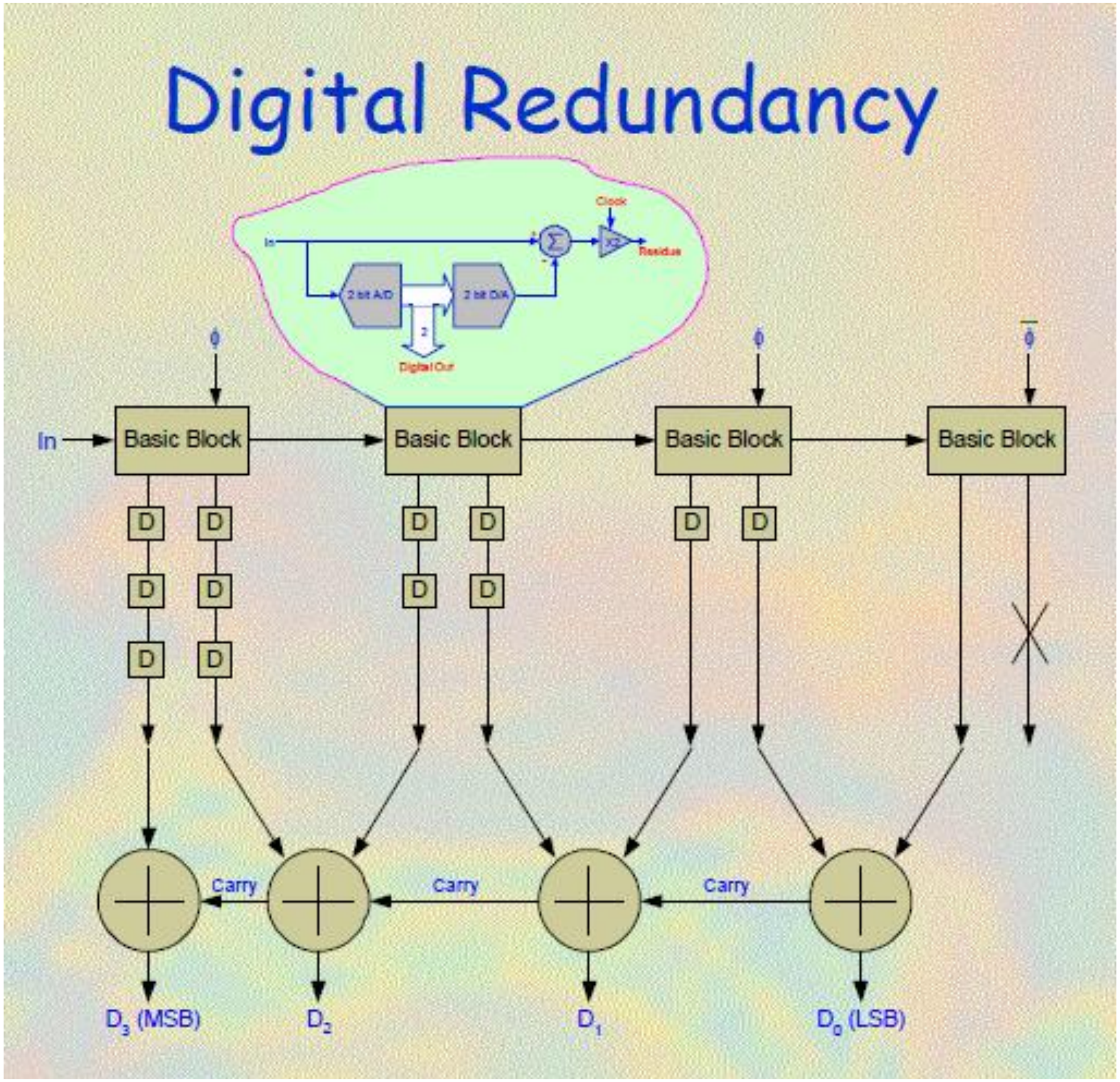
Most modern pipelined ADCs employ a technique called "digital error correction" to greatly reduce the accuracy requirement of the flash ADCs. In Figure , notice that the 3-bit residue at the summation-node output has a dynamic range one-eighth that of the original Stage 1 input (VIN), yet the subsequent gain is only 4. Therefore, the input to Stage 2 occupies only half the range of the 3-bit ADC in Stage 2 (that is, when there is no error in the first 3-bit conversion in Stage 1).

If one of the comparators in the first 3-bit flash ADC has a significant offset when an analog input is applied, then an incorrect 3-bit code and thus an incorrect 3-bit DAC output would result, thus producing a different residue. As long as this gained-up residue does not overrange the subsequent 3-bit ADC, it can be proven that the LSB code generated by the remaining pipeline (when added to the incorrect 3-bit MSB code) will give the correct ADC output code..

The digital error correction will not correct for errors made in the final 4-bit flash conversion. Any error made at that conversion is suppressed by the large (4exp4) cumulative gain preceding the 4-bit flash. Thus the final stage only needs to be more than 4-bits accurate.

Although each stage generates three raw bits in the Figure example, because the interstage gain is only 4, each stage (Stages 1 to 4) effectively resolves only two bits. The extra bit is simply to reduce the size of the residue by one half, allowing extra range in the next 3-bit ADC for digital error correction, as mentioned above. This process is called "1-bit overlap" between adjacent stages. The effective number of bits of the entire ADC is therefore 2 + 2 + 2 + 2 + 4 = 12 bits.

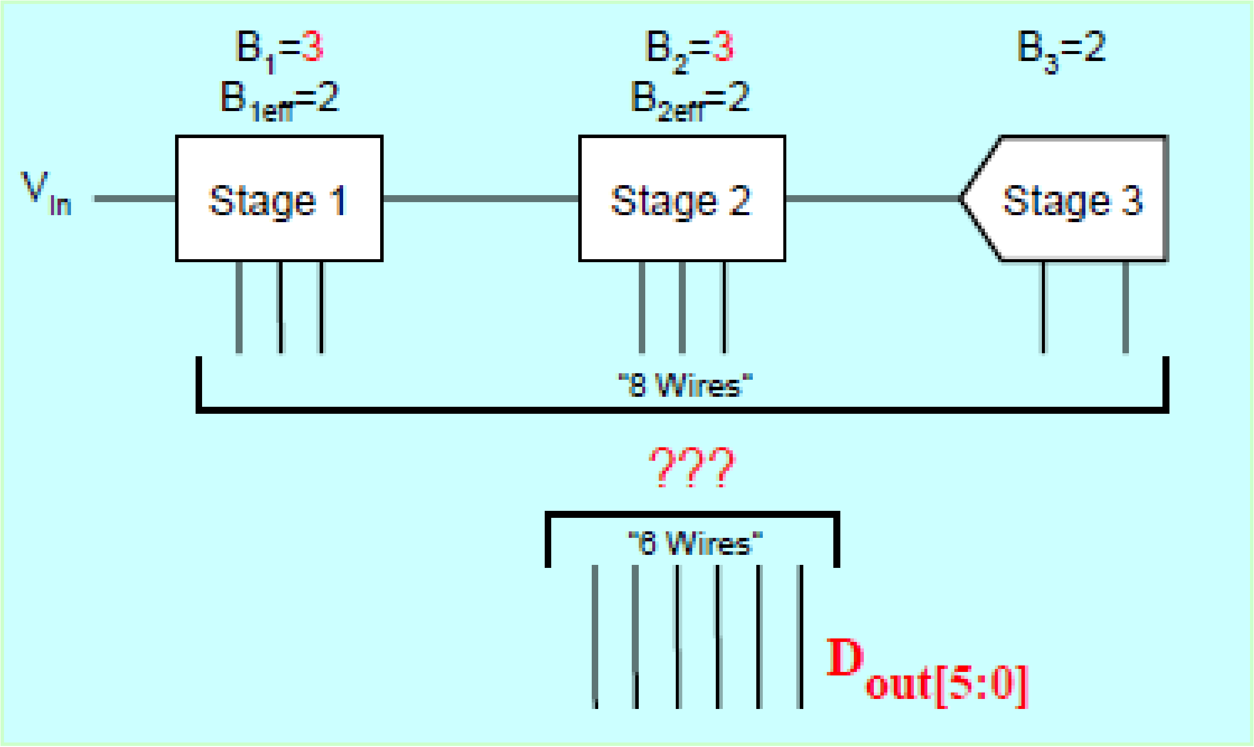
Digital Error Correction –Bits combination



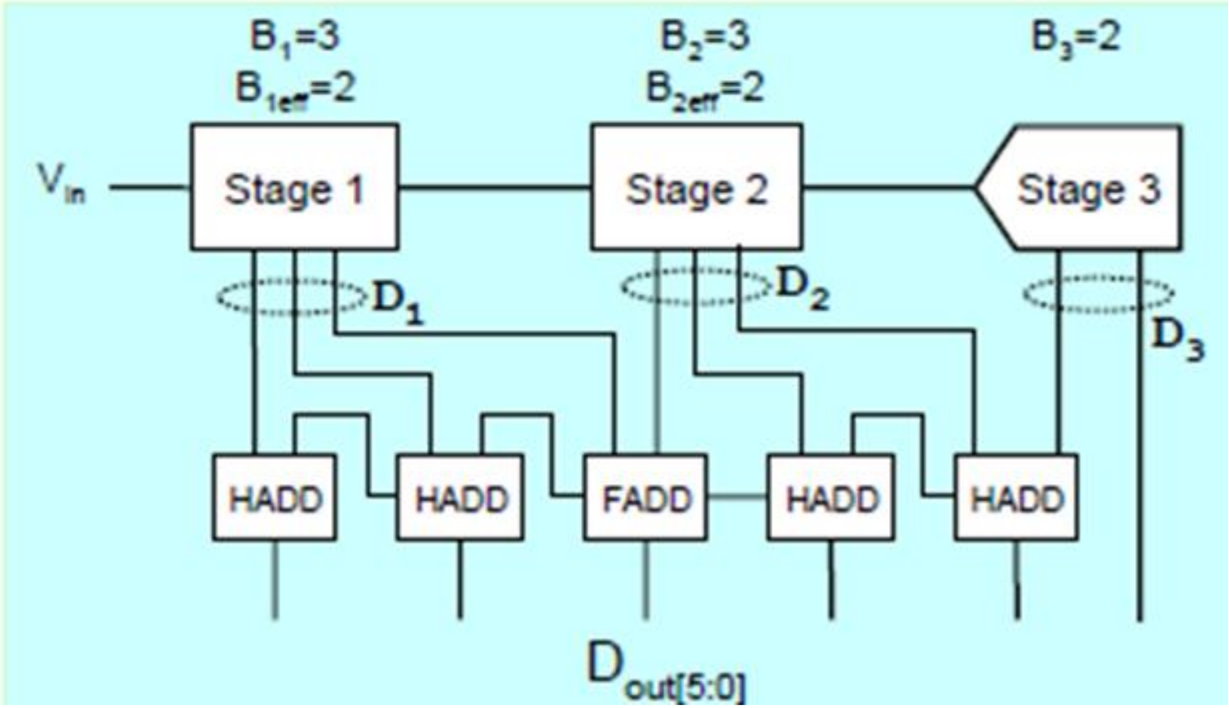
Digital Error Correction –Bits combination 6-Bits 3-Stage pipeline ADC example

Combining the Bits Including Redundancy

- Example: Three 2-bit stages, incorporating 1- bit redundancy in stages 1 and 2



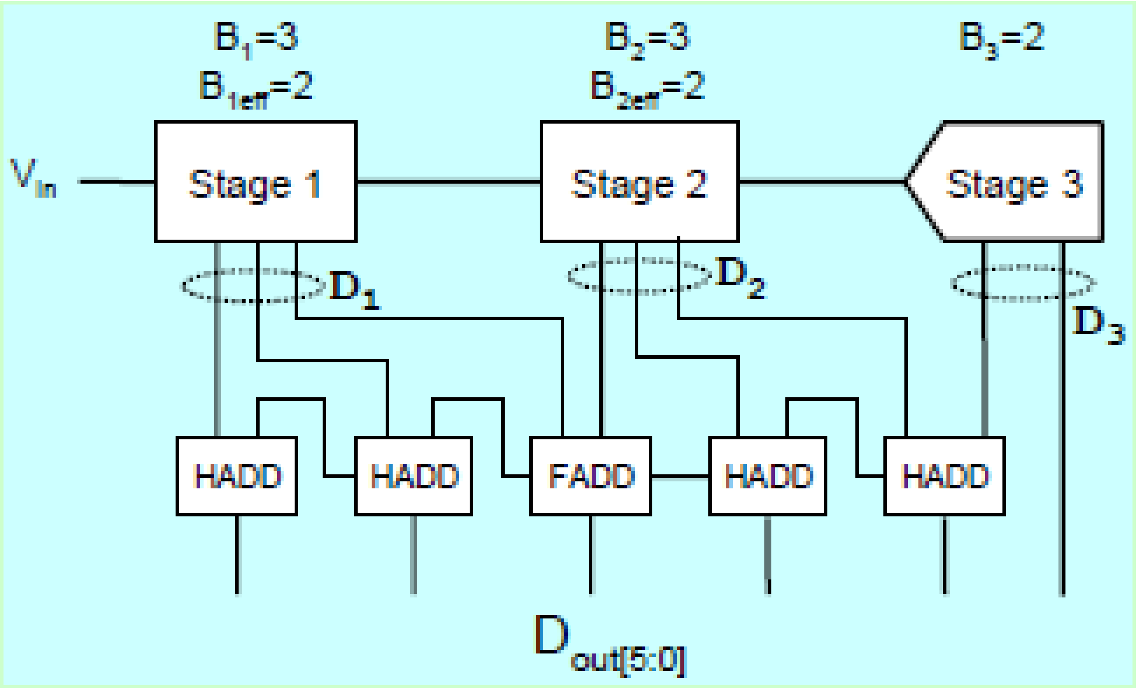
Combining the Bits



- Bits overlap
- Need adders

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

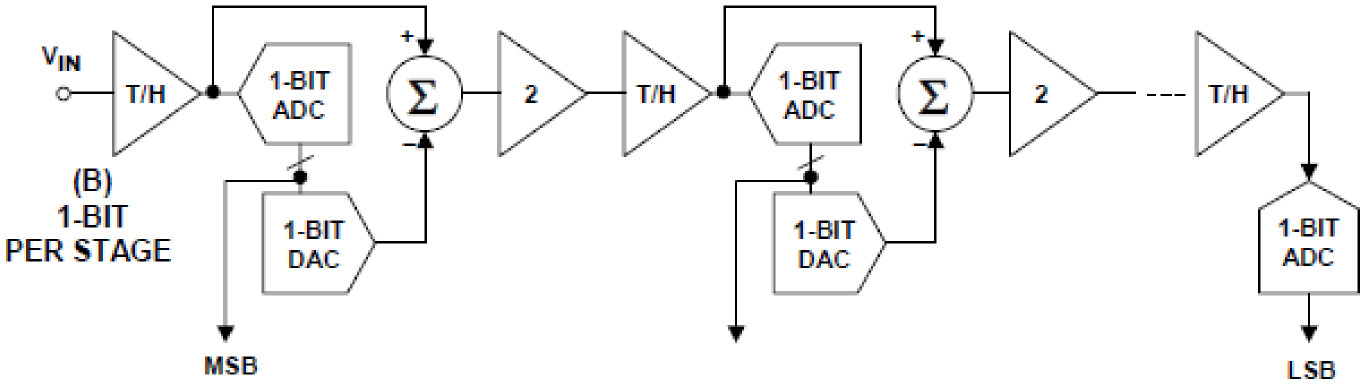
Combining the Bits Example



```
D1    001
D2      111
D3        10
--------------
Dout  011000
```

Fully Pipelined ADCs (with identical stages)

Fully Pipelined ADC with Identical 1-bit Stages

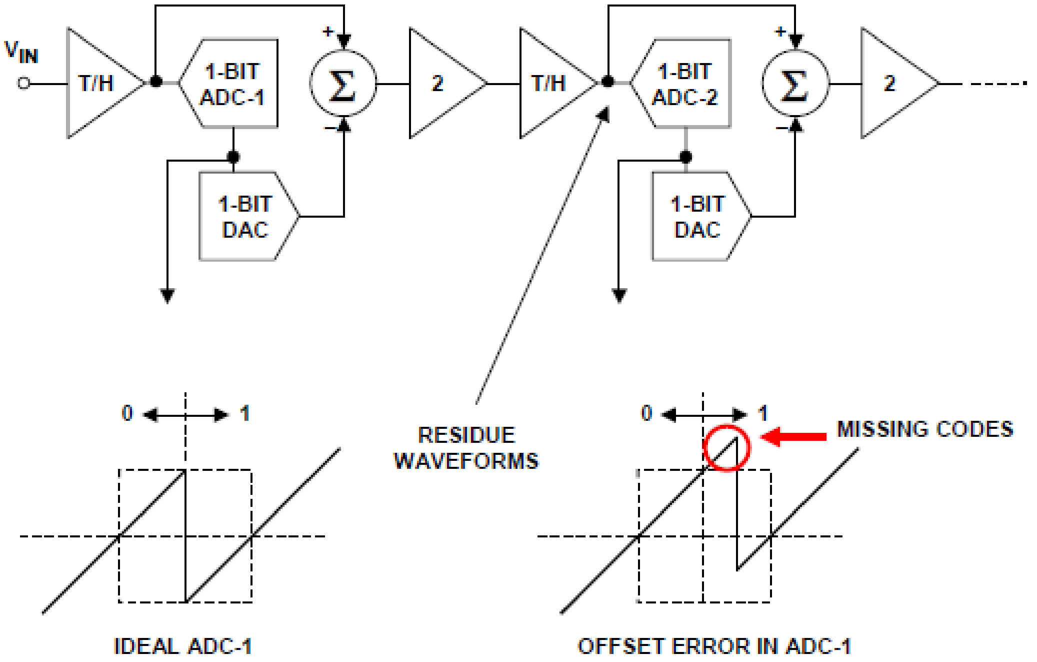


Higher speed CMOS pipelined ADCs tend to favor a lower number of bits per stage ,as low as just one bit per stage so that the interstage gain is only 2, because it is difficult to realize wideband amplifiers of very high gain in CMOS.
Lower sampling-rate CMOS pipelined ADCs and bipolar pipelined ADCs (even those with a very high sampling rate) tend to favor more bits per stage. This also results in less data latency

Advanced CMOS family 1-bit pipelined ADCs (e.g. 10-bit, 20Msps and 10-bit, 10Msps ADCs) uses the popular 1.5-bit-per-stage architecture; each stage resolves one bit with 0.5-bit overlap. Each 1.5-bit stage has a 1.5-bit flash ADC (only two comparators), versus a full 2-bit flash ADC. It can be shown that, with digital error correction, this works the same way as a regular pipelined ADC with 2-bit flash ADC and DAC. These converters achieve a high SNR of 59dB with 10MHz analog inputs sampled at 20Msps.

1.5-Bit Stages Architecture (in pipelined ADCs)

Error correction is used in practically all pipelined ADCs, including the simple 1-bit stage. Figure below shows how an ADC constructed of uncorrected cascaded 1-bit stages will ultimately result in missing codes unless each stage is nearly ideal.

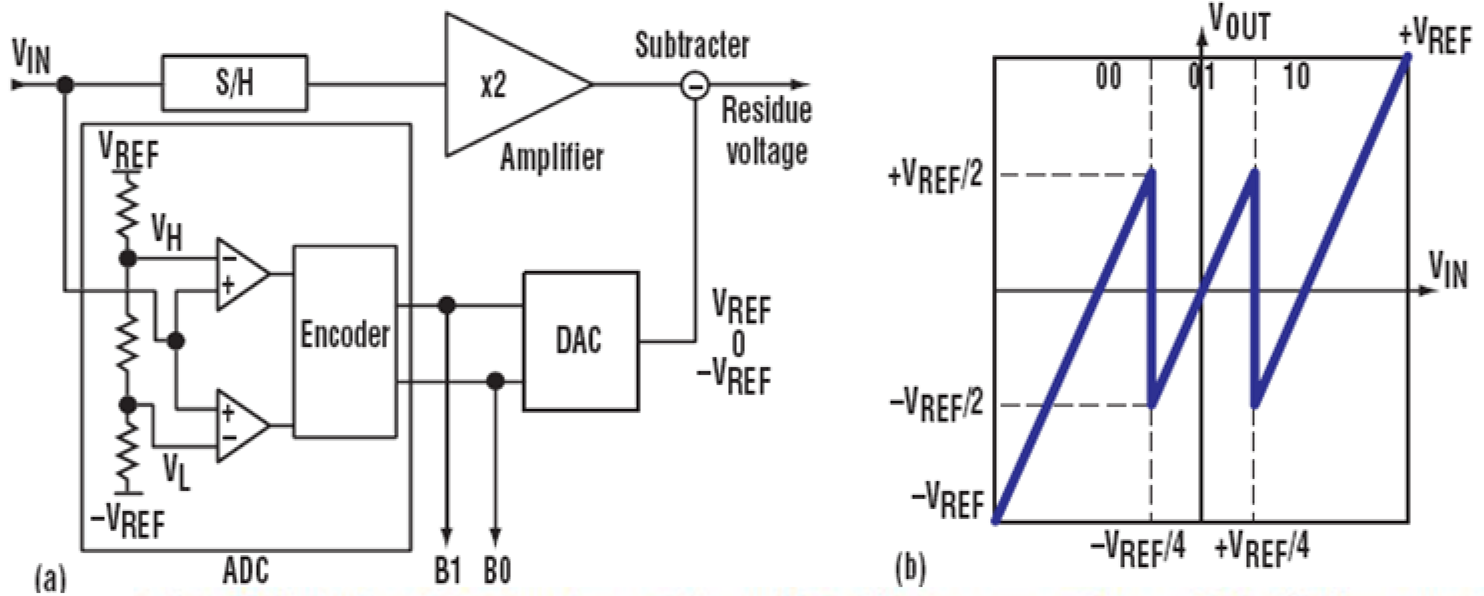


Error correction can be added to the simple 1-bit stage by adding a single extra comparator—resulting in what is commonly referred to as a "1.5-bit" stage

The two comparators have three possible output codes: 00, 01, and 10. Note that three parallel comparators form a complete 2-bit stage—which would be required for the final stage in a pipelined 1.5-bit ADC, as one additional output level is required to generate the 11 code.

1.5-Bit Stages Architecture (in pipelined ADCs)

Basic Structure of 1.5 bit stage A 1.5-bit stage is a 1-bit stage into which some redundancy is built to provide a large tolerance for component tolerances and imperfections. A digital correction algorithm later eliminates the redundancy. A 1.5-bit stage is actually a stage that represents approximately 1.5 bits.

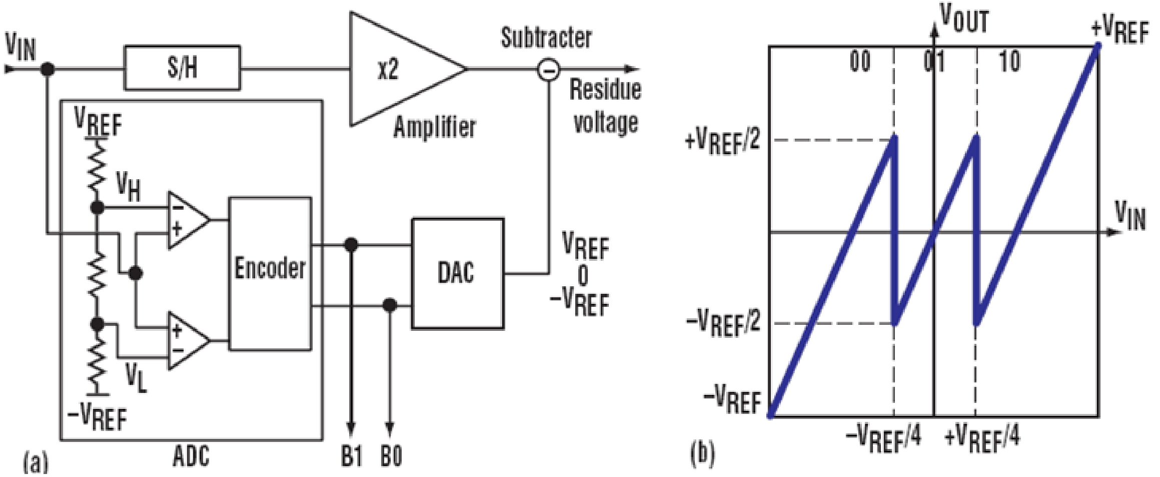


2. A 1.5-bit pipeline ADC stage has a S/H, an ADC with two comparators, a DAC with three possible output voltages, a subtracter, and a x2 amplifier. The stage voltage transfer function is highly nonlinear.

The 1.5-bit stage uses two symmetrical analog comparison levels, VH and VL, instead of a single level in 1 bit/stage. The amplifier has a gain of 2. Choice of voltage levels VH and VL isn't critical, but they are usually set at $V_H = 0.25\ V_{REF}$ and VL = -0.25 VREF

The operating voltage range is divided into three sections: High (H) above V_H, Mid (M) between V_H and VL, and Low (L) negative of V_L. This system is known as Redundant Signed Digit (RDS)

1.5-Bit Stages Architecture (in pipelined ADCs)



The low-resolution ADC stage comprises two comparators plus some simple encoding. The ADC output consists of two bits—B1 and B0. This is the initial digital output, before code conversion and error correction. The output codes are 00, 01, and 10 for V_{IN} in the L, M, and H input ranges, respectively. The DAC outputs are -V_{REF}, 0, and +V_{REF} for V_{IN} in the L, M, and H input ranges. The analog residue voltages out of the stage after subtraction along with aforementioned parameters are shown in following Table

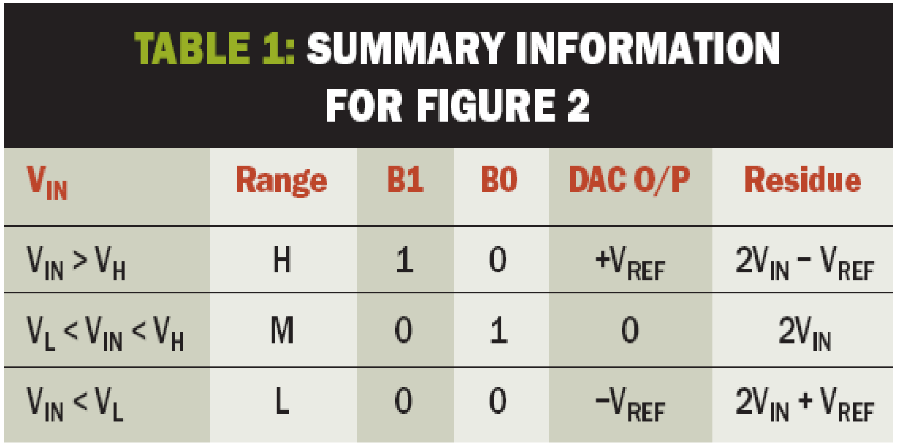


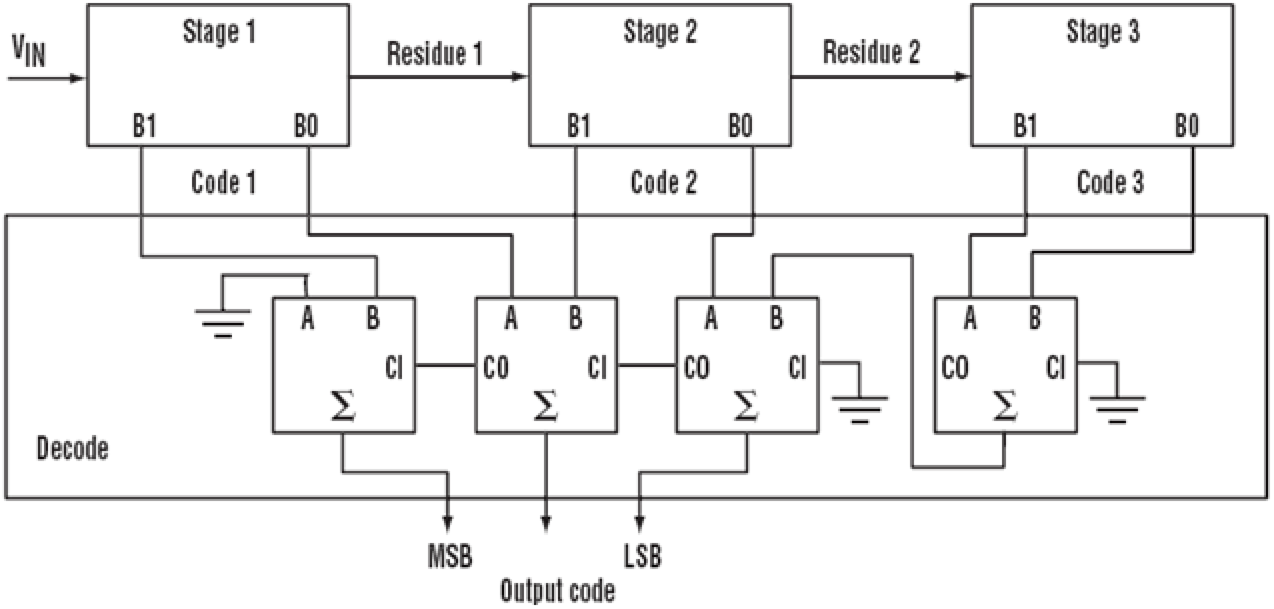
TABLE 1: SUMMARY INFORMATION FOR FIGURE 2

V_{IN}	Range	B1	B0	DAC O/P	Residue
$V_{IN} > V_H$	H	1	0	$+V_{REF}$	$2V_{IN} - V_{REF}$
$V_L < V_{IN} < V_H$	M	0	1	0	$2V_{IN}$
$V_{IN} < V_L$	L	0	0	$-V_{REF}$	$2V_{IN} + V_{REF}$

1.5-Bit Stages Architecture (in pipelined ADCs)

Each 1.5-bit pipelined ADC stage, as described earlier, produces a 2-bit code. Once the error-correction algorithm is applied, this is reduced to the final one-bit-per-stage code. Even in the absence of any errors at all, the 2-bit-per-stage code must be converted to 1 bit per stage.

An example follows illustrating these issues, including error-correction code conversion. Figure below shows a three-stage cascade of 1.5-bit stages in block diagram form, accompanied by _Table 2



3. In this example, each of three cascaded 1.5-bit pipeline ADC stages outputs a 2-bit code, which after code conversion done with adders results in the final 3-bit output code.

1.5-Bit Stages Architecture (in pipelined ADCs)

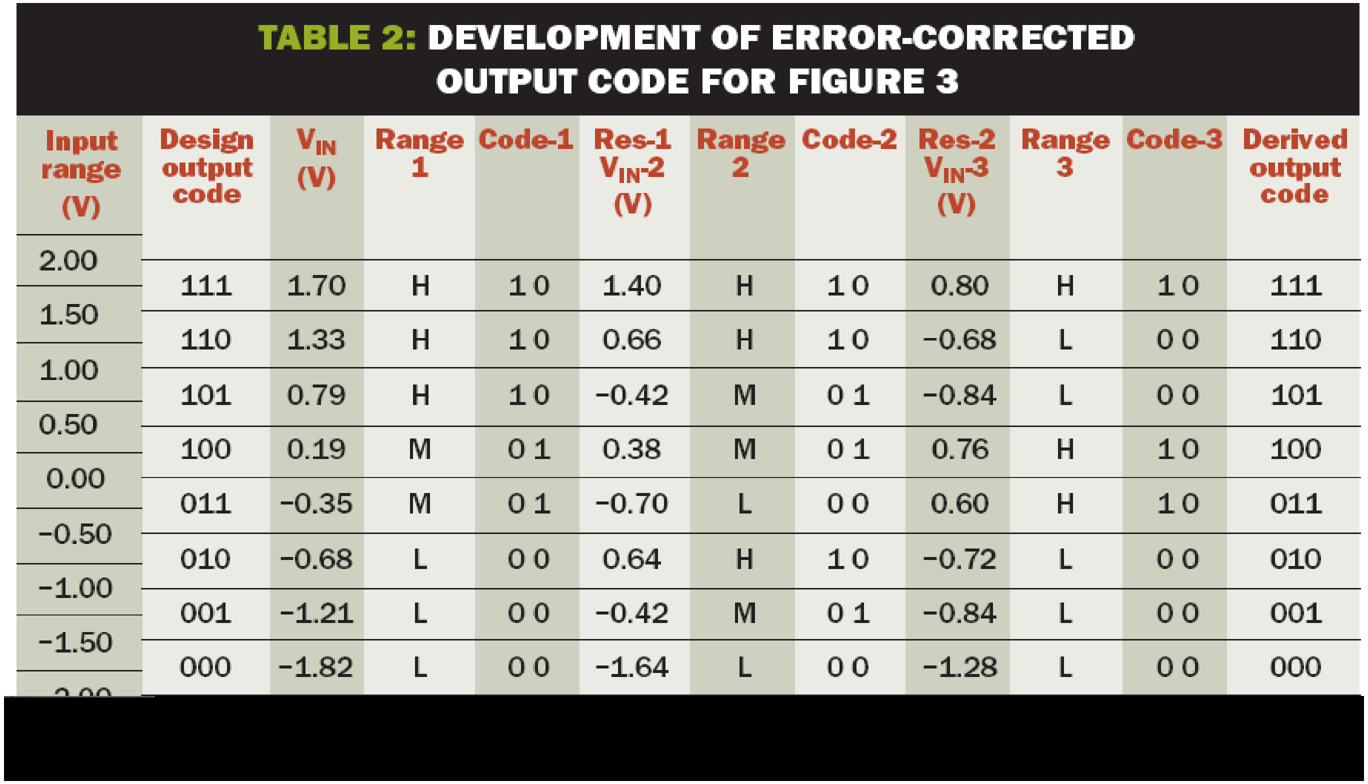


TABLE 2: DEVELOPMENT OF ERROR-CORRECTED OUTPUT CODE FOR FIGURE 3

Input range (V)	Design output code	V_{IN} (V)	Range 1	Code-1	Res-1 V_{IN}-2 (V)	Range 2	Code-2	Res-2 V_{IN}-3 (V)	Range 3	Code-3	Derived output code
2.00	111	1.70	H	1 0	1.40	H	1 0	0.80	H	1 0	111
1.50	110	1.33	H	1 0	0.66	H	1 0	−0.68	L	0 0	110
1.00	101	0.79	H	1 0	−0.42	M	0 1	−0.84	L	0 0	101
0.50	100	0.19	M	0 1	0.38	M	0 1	0.76	H	1 0	100
0.00	011	−0.35	M	0 1	−0.70	L	0 0	0.60	H	1 0	011
−0.50	010	−0.68	L	0 0	0.64	H	1 0	−0.72	L	0 0	010
−1.00	001	−1.21	L	0 0	−0.42	M	0 1	−0.84	L	0 0	001
−1.50	000	−1.82	L	0 0	−1.64	L	0 0	−1.28	L	0 0	000

[redacted]

Because the resolution is 3 bits, the input voltage range of ±2 V is divided into eight equal sectors as shown in the first column of Table 2. The intended output code is binary increasing from negative to positive, as shown in the second column, which is labeled design output code.

The third column of Table 2 lists arbitrarily selected input voltages, one in each of the eight equal input-voltage sectors. The residue voltages of the first two stages are shown, as are the three sets of two-digit uncorrected output codes from each of the three stages.

1.5-Bit Stages Architecture (in pipelined ADCs)

Calculation Example of the final output code

To generate the final code-converted and error-corrected 3-bit output code from the three 2-bit stage codes, the 2-bit digital outputs from each stage are added together with 1 bit overlapped between adjacent stages. The three MSBs are the final code.

E.g. for the VIN = 0.79 V example, the output codes from the three stages are 10, 01, 00. The final 3-bit output code is obtained as follows:

$$\begin{array}{cccc}
1 & 0 & & \\
 & 0 & 1 & \\
 & & 0 & 0 \\
\hline
1 & 0 & 1 & 0
\end{array}$$

Ignoring the far-right digit, the final output code is 1 0 1.

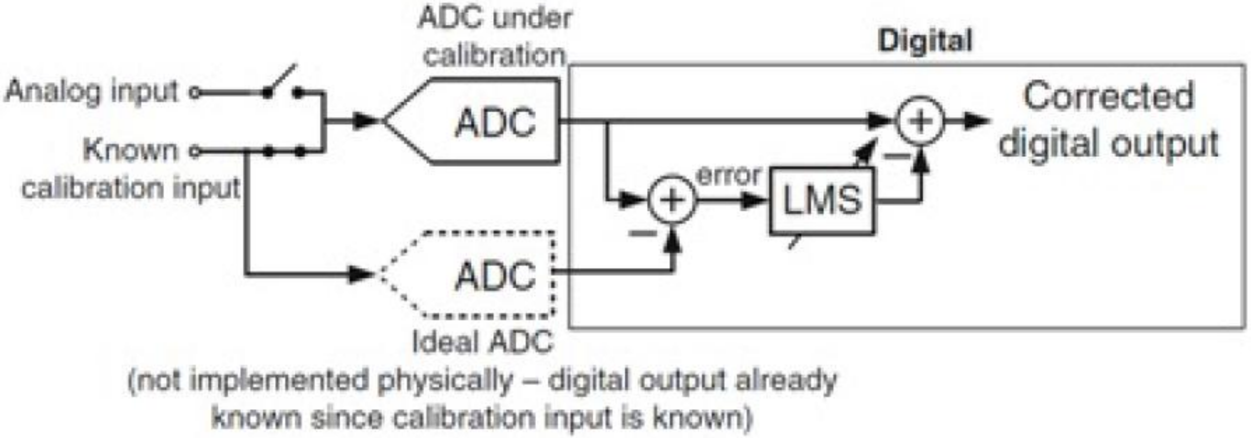
Calibration Methods in Pipelined ADCs

Beyond the bit redundancy that it is usually necessary to be employed in each stage of pipelined ADCs (as explained in previous slides) there is also need for calibration in order to try to reduce mismatches, offsets and non-linearities so that in combination with the bit redundancy technique higher quality outputs could be derived.

Some indicative such calibration methods are described in the following slides

Foreground Calibration

In a foreground calibration scheme, the unknown errors are estimated by interrupting the operation of the ADC and then injecting a known signal. The expected output is compared to the actual output to measure the error. Once the error is acquired, Least Mean Square (LMS) algorithms can be used to correct for the error.

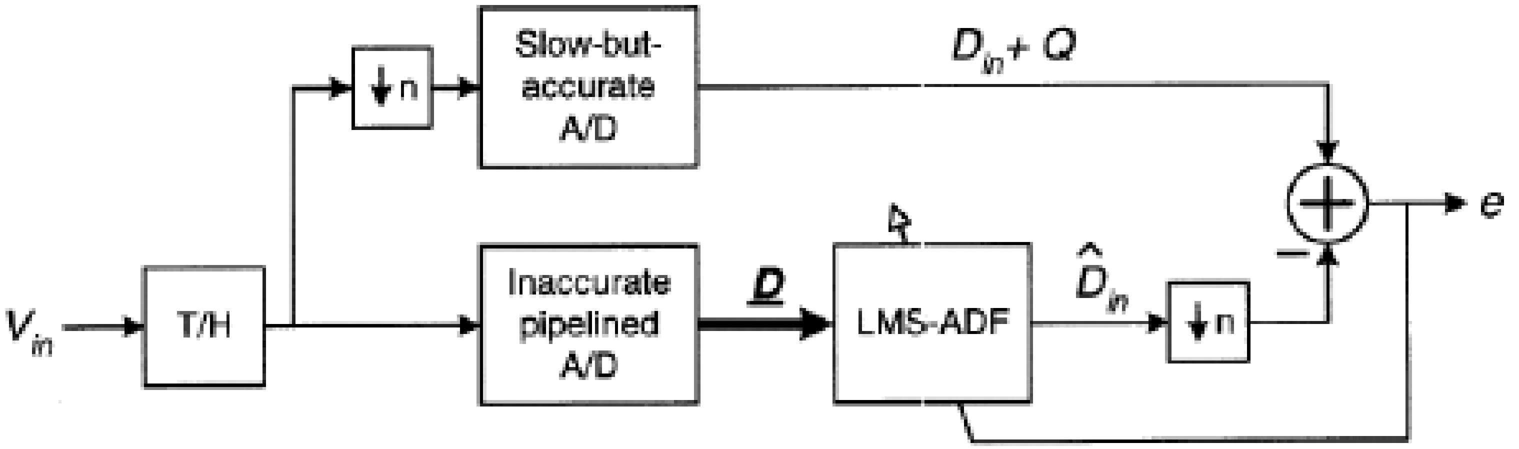


As shown in Figure, analog input signal is fed into the actual ADC and a known signal is fed into the ideal ADC. Since it is impossible to implement an ideal ADC, this component is simulated digitally. Another digital component is used to calculate the error between the actual output and the ideal output. This same digital component will then correct the digital output for this calculated error. The main advantage of using foreground calibration is that one can achieve the corrected digital output in a few clock cycles. However, the operation of the ADC is interrupted during calibration. This interruption is impractical in some applications

Background Calibration

Background calibration technology can correct errors of ADC circuits without interrupting the operation of the ADC. Methods of background calibration can be analog or digital and have a variety of implementations.

Least Mean Square Adaptive Digital Background Calibration of Pipelined Analog-to-Digital Converters



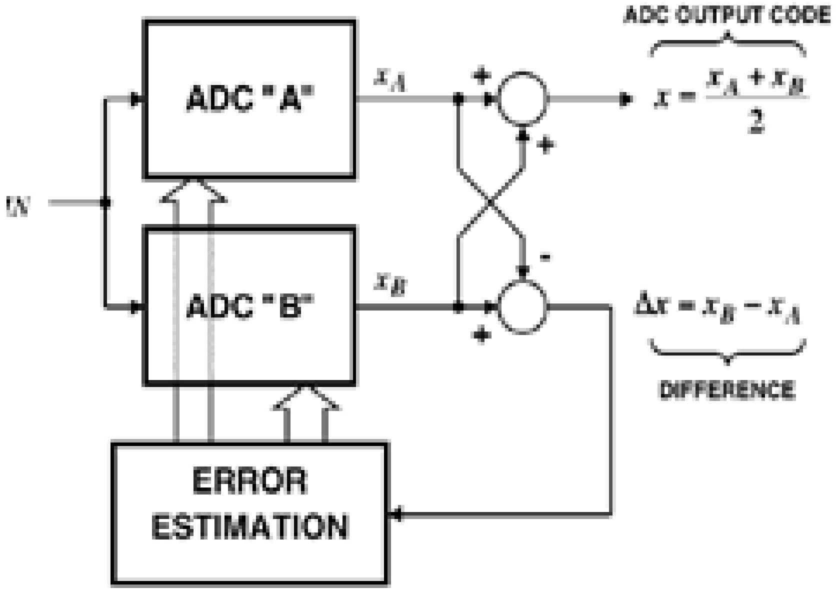
- Slow, but accurate ADC operates in parallel with pipelined (main) ADC
- Slow ADC samples input signal at a lower sampling rate (f_s/n)
- Difference between corresponding samples for two ADCs (e) used to correct fast ADC digital output via an adaptive digital filter (ADF) based on minimizing the <u>L</u>east-<u>M</u>ean-<u>S</u>quared error

Ref: Y. Chiu, *et al*, "Least Mean Square Adaptive Digital Background Calibration of Pipelined Analog-to-Digital Converters," IEEE TRANS. CAS, VOL. 51, NO. 1, JANUARY 2004

Split ADC Architecture

The split ADC architecture is known for being able to calibrate residue gain error over a short period of time .It can also digitally correct DAC errors in pipeline ADCs. In the split ADC architecture, there are two ADCs with the same resolution. The only difference between them is the residue transfer characteristic. Those two ADCs are placed in parallel and are applied with the same input signal.

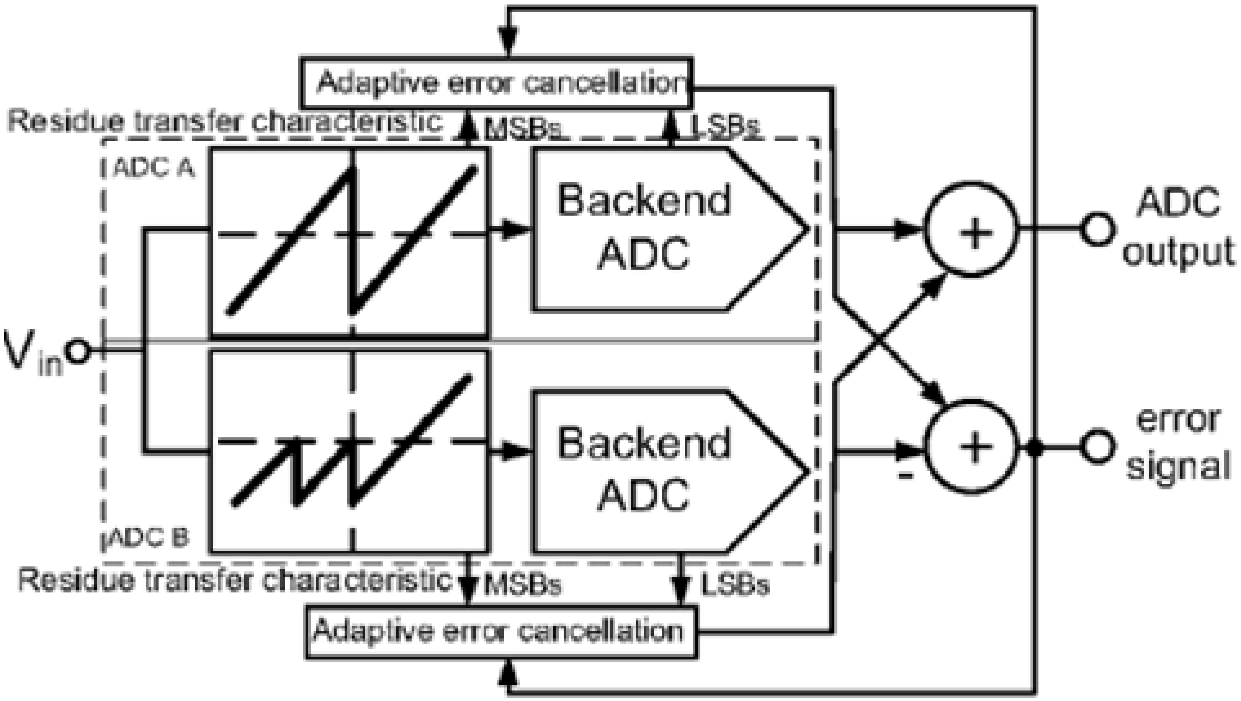
A split pipeline ADC architecture is shown at figure on the left . The ADC is split into two identical ADCs, processing the same input but producing different outputs. The average of the two outputs becomes the output of the ADC. The difference between the two outputs is used to calibrate the ADC. If the difference between two outputs is zero, there is no error and the ADC is calibrated perfectly. If not, that difference is used to adapt the error corrective term and update the calibration parameters in each ADC to achieve an error of zero.



Example:In a 12-bit pipeline ADC, the authors incorporated two stages in each of the split ADCs in their design. The 1st stage consists of a 4-bit pipeline stage and the second stage consists of a single 10-bit flash ADC. In this work, only the 1st stage is calibrated and the second stage does not need to be calibrated. Even though the goal is to implement a 12-bit ADC, they included two extra bits to achieve more accuracy in error correction.

Split ADC Architecture

Block Diagram Detail


Adaptive error cancellation
Residue transfer characteristic
MSBs
LSBs
ADC A
Backend ADC
ADC output
V_{in}
Backend ADC
error signal
-
ADC B
Residue transfer characteristic
MSBs
LSBs
Adaptive error cancellation

Pipelined ADCs –Pros and Cons

Pipeline ADC architectures simplify ADC design and provide other advantages :

•Extra bits per stage optimize correction for overlapping errors.

•Separate track-and-hold (T/H) amplifiers for each stage release each previous T/H to process the next incoming sample, enabling conversion of multiple samples simultaneously in different stages of the pipeline.

•Lower power consumption.

•Higher-speed ADCs (fCONV > 100ns, typical) entail less cost and less design time and effort.

•Fewer comparators to become metastable virtually eliminates sparkle codes and thermometer bubbles.

•Pipelined ADCs are available today with resolutions =14 bits and rates > 100 MHz.

Applications

Pipeline converters fit high-speed applications (5 MHz to >100MHz). They are ideal for many applications that require not only high sampling rates but high signal-to-noise ratio (SNR) and spurious-free dynamic range (SFDR). A popular application for these converters today is in software-defined radios (SDR) that are used in modern cellular telephone base stations.

• Applications where you typically find pipeline converters are:

– Wireless and Line Communications, Test and Measurement, Instrumentation,– Medical Imaging, – Radar Systems, – Data Acquisition

Pipelined ADCs –Pros and Cons

But pipeline ADCs also impose difficulties:

•Complex reference circuitry and biasing schemes.

•Pipeline latency, caused by the number of stages through which the input signal must pass-This latency might be a problem in some applications. If the ADC is within a feedback control loop, latency may be a problem. Latency also makes pipelined ADCs difficult to use in multiplexed applications.

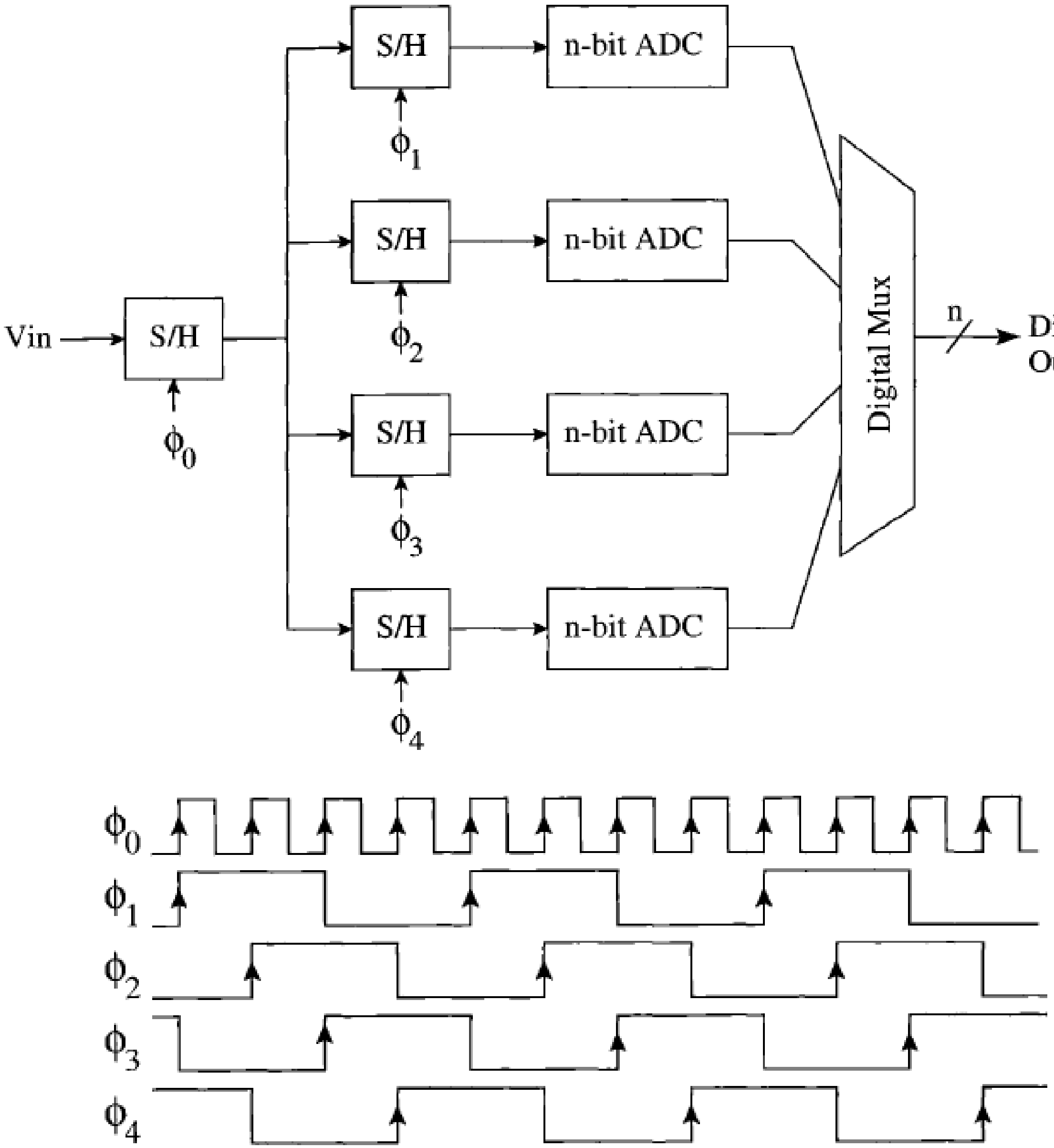
•Obviously, this precludes operation in single-shot or burst-mode applications—where the SAR ADC architecture is more appropriate

•Critical latch timing, needed for synchronization of all outputs.

•Sensitivity to process imperfections that cause nonlinearities in gain, offset, and other parameters-Greater sensitivity to board layout, compared with other architectures.

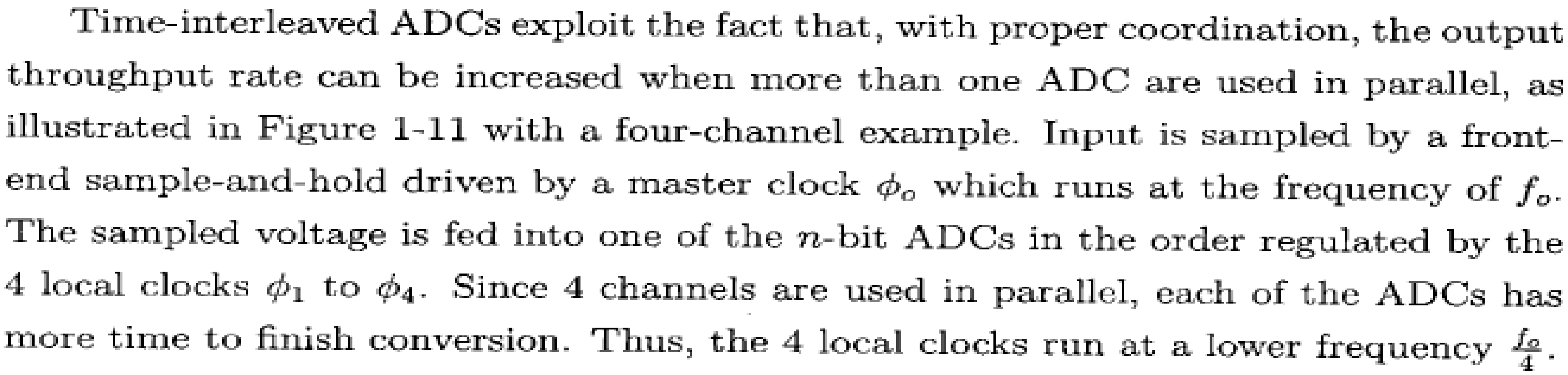
•An issue exists relating to most CMOS pipelined ADCs is their performance at low sampling rates: Very low sampling rates extend the hold times for the internal track-and-holds to the point where excessive droop causes conversion errors. Therefore, most pipelined ADCs have a specification for minimum as well as maximum sampling rate.

Time Interleaved ADCs

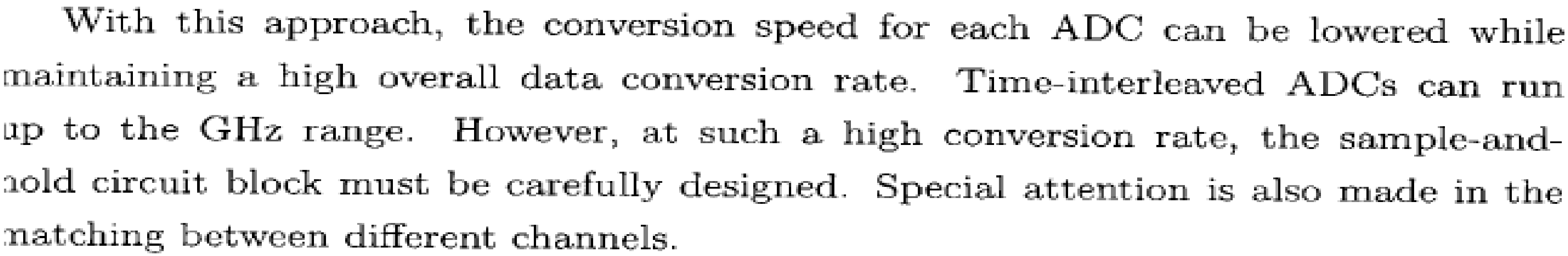


Time Interleaved ADCs

Basic Mechanism



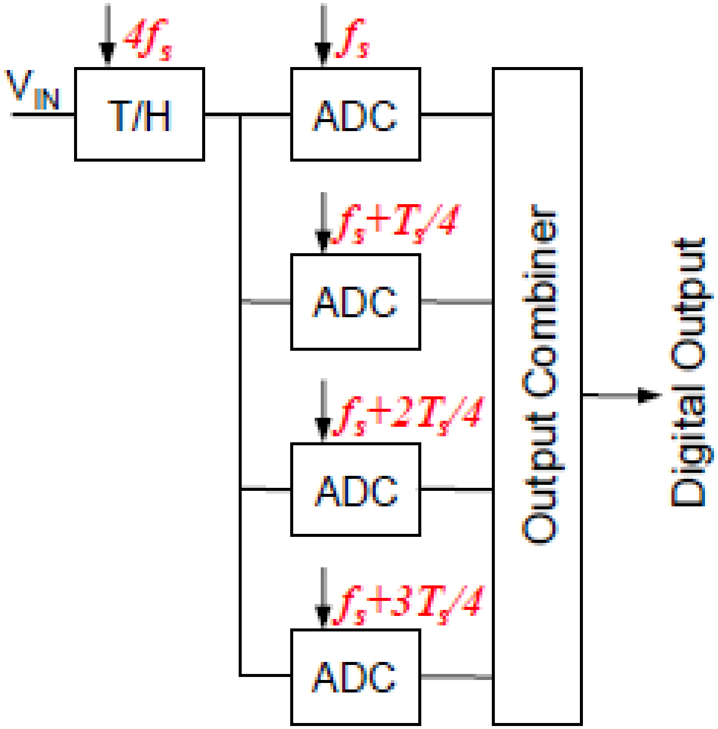
Time-interleaved ADCs exploit the fact that, with proper coordination, the output throughput rate can be increased when more than one ADC are used in parallel, as illustrated in Figure 1-11 with a four-channel example. Input is sampled by a front-end sample-and-hold driven by a master clock ϕ_o which runs at the frequency of f_o. The sampled voltage is fed into one of the n-bit ADCs in the order regulated by the 4 local clocks ϕ_1 to ϕ_4. Since 4 channels are used in parallel, each of the ADCs has more time to finish conversion. Thus, the 4 local clocks run at a lower frequency $\frac{f_o}{4}$.



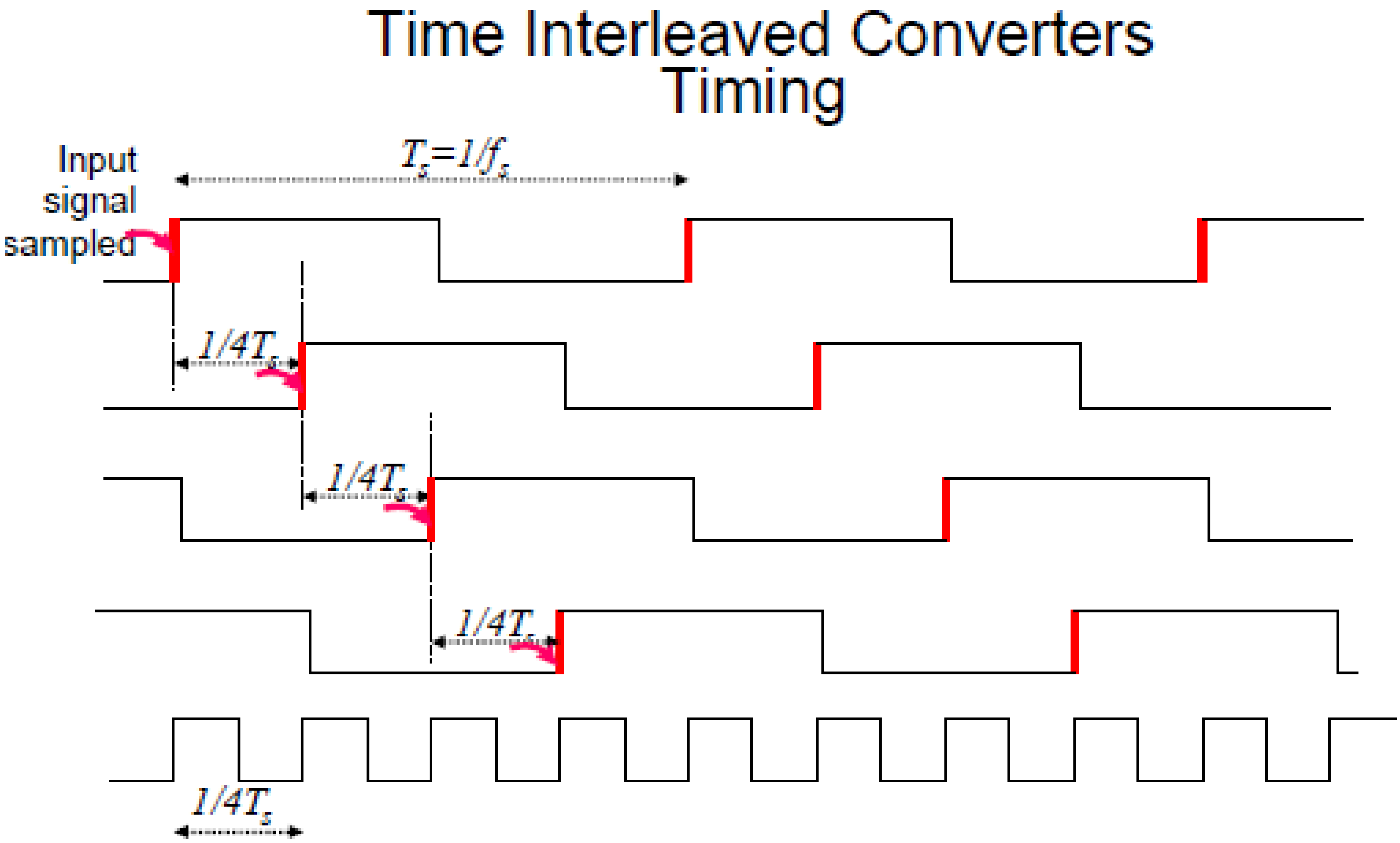
With this approach, the conversion speed for each ADC can be lowered while maintaining a high overall data conversion rate. Time-interleaved ADCs can run up to the GHz range. However, at such a high conversion rate, the sample-and-hold circuit block must be carefully designed. Special attention is also made in the matching between different channels.

Time Interleaved ADCS- 4 ADCs Example

- Example:
  - 4 ADCs operating in parallel at sampling frequency f_s
  - Each ADC converts on one of the 4 possible clock phases
  - Overall sampling frequency= $4f_s$
  - Note T/H has to operate at $4f_s$!
- Extremely fast:
  Typically, limited by speed of T/H
- Accuracy limited by mismatch among individual ADCs (timing, offset, gain, …)



Time Interleaved ADCS- 4 ADCs Example



- Note: Effective sampling rate→ $4xf_s$

Time Interleaved ADCs

Discussion-Pros and Cons

Theoretically, the conversion rate can be increased by the number of parallel paths, at the cost of a linear increase in power and chip area. This time-interleaving architecture has the following major sources of distortion.

One error source is that a timing mismatch among the input samplers of each channel can degrade spectrum purity- unavoidable because of asymmetry among the clock distribution in the layout, and also due to mismatch of devices such as clock buffer devices.

The other sources of distortions are the offset and gain mismatch among these channels. The inter-channel offset mismatch gives rise to fixed pattern noise (distortion). This can be found in the frequency domain as a tone at multiples of fs/N, N=number of channels and n=1, 2,…, N.

Folding ADCs

Why Folded ADCs?

The fastest architectures for A/D conversion are the full flash ADC in which the whole A/D conversion finishes in one single step and the pipeline ADC which, after an initial delay of N clock cycles, conversion is accomplished in one clock cycle.

The full flash ADC, however, suffers from large die area when the resolution is greater than 6-8 bits. The number of the comparators needed in full flash ADC explodes exponentially with the resolution.

The two-step or multiple-step ADCs require much fewer comparators than flash ADC but need two or several steps to finish conversion and are therefore slower.

Although the pipeline technique can be used to improve the throughput in multiple-step ADCs, they still pose a high initial latency as well as overhead of the TH (Track and Hold) circuitry needed between the stages. which - makes their implementation more challenging.

A number of circuit architectures have been developed to alleviate the area problem while maintaining the one-step conversion. Among them is the folding ADC

Folding ADCs-Similarity to 2-stepADCs

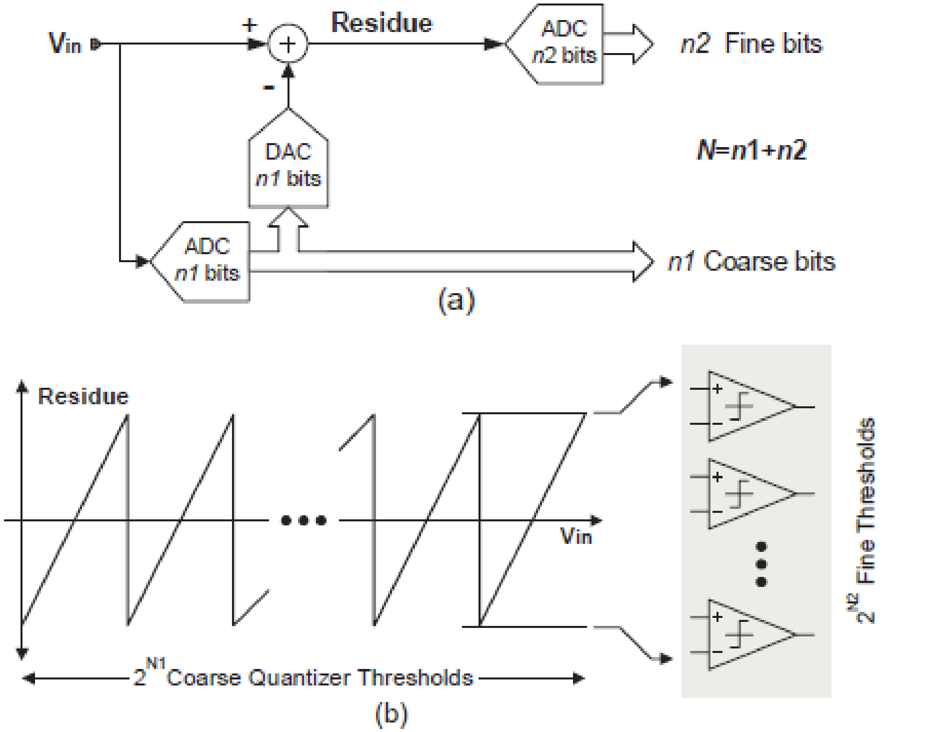
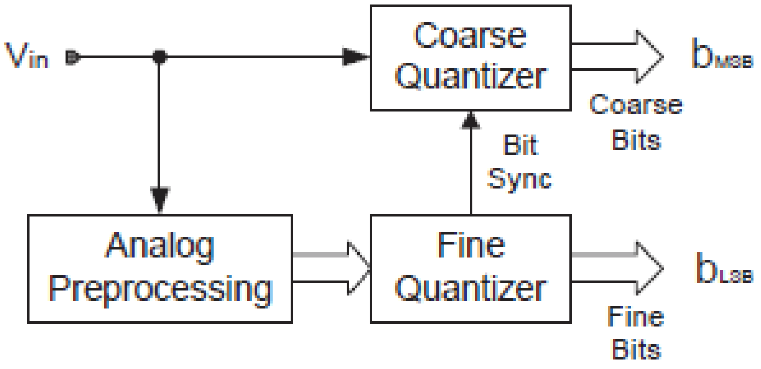


Figure 25 Architecture of two-step A/D converter (a) block diagram (b) basic principle

A two-step A/D converter gains efficiency by partitioning an *N-bit quantization into two lower-resolution quantizations. In such a converter* (Figure 25a) an *n1-bit coarse quantizer digitizes the input signal with low resolution, and* applies the resultant codeword to reconstruction DAC.

The analog output of the DAC is then subtracted from the original input to form a residue signal (Figure 25b), which is quantized by an *n2-bit quantizer. The advantage of this approach arises because the* combined complexity of the *n1-bit coarse quantizer and the n2-bit fine quantizer can be* far less than the complexity of a single *N-bit quantizer.*

Folding ADCs-Similarity to 2-stepADCs

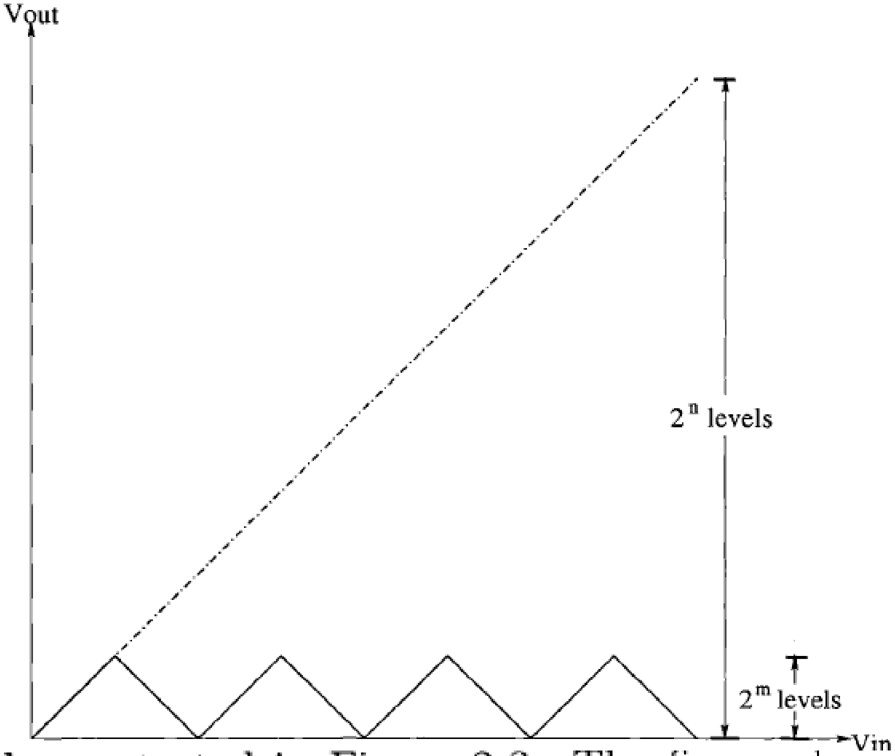


The idea of folding is similar to a two-step ADC: both structures utilize two lower resolution quantizers to implement one higher resolution ADC.

However, folding ADCs use analog preprocessing to generate the “residue” signal at the same time instant when the MSBs from the coarse quantizer are produced. Also the coarse quantizer determines where the input lies for the folding amplifier (analog preprocessing).

The total resolution of the folding ADC is $N_B = n_{MSB} + n_{LSB}$, *where* n_{MSB} *and* n_{LSB} *are the numbers of bits resolved* in the coarse and fine quantizers, respectively.

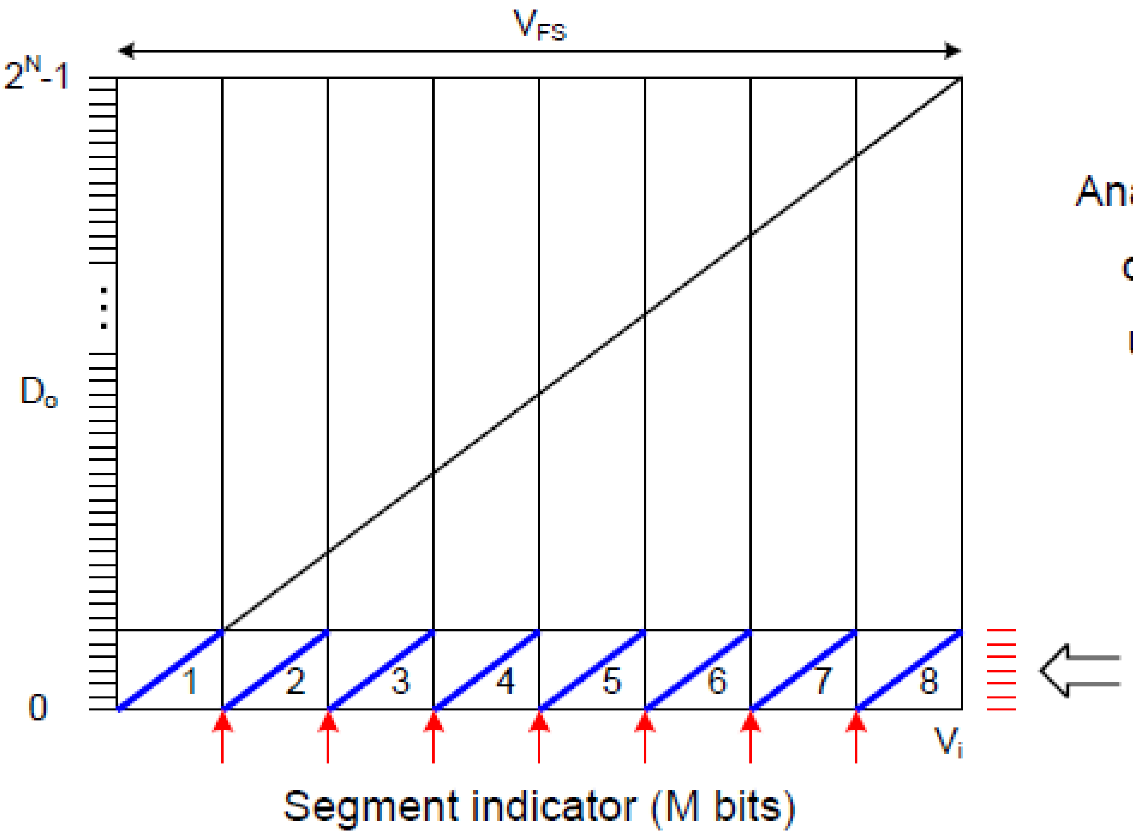
Folding ADCs-Folding Mechanism



The idea is demonstrated in Figure 2-2. The figure shows both the transfer curves of an unfolded and folded signal. Without folding, v_{out} spans across a much wider range, so more comparators have to be used in order to achieve conversion at a given accuracy. However, the folding amplifier folds the signal into a triangular waveform, so the range of v_{out} is reduced, as well as the number of comparators required to obtain the same accuracy as before. Since the folding operation maps more than one input value v_{in} to the same output value v_{out}, the coarse ADC is responsible for determining in which fold the input lies. This process is done in parallel with the fine ADC conversion.

Folding ADCs-Folding Mechanism

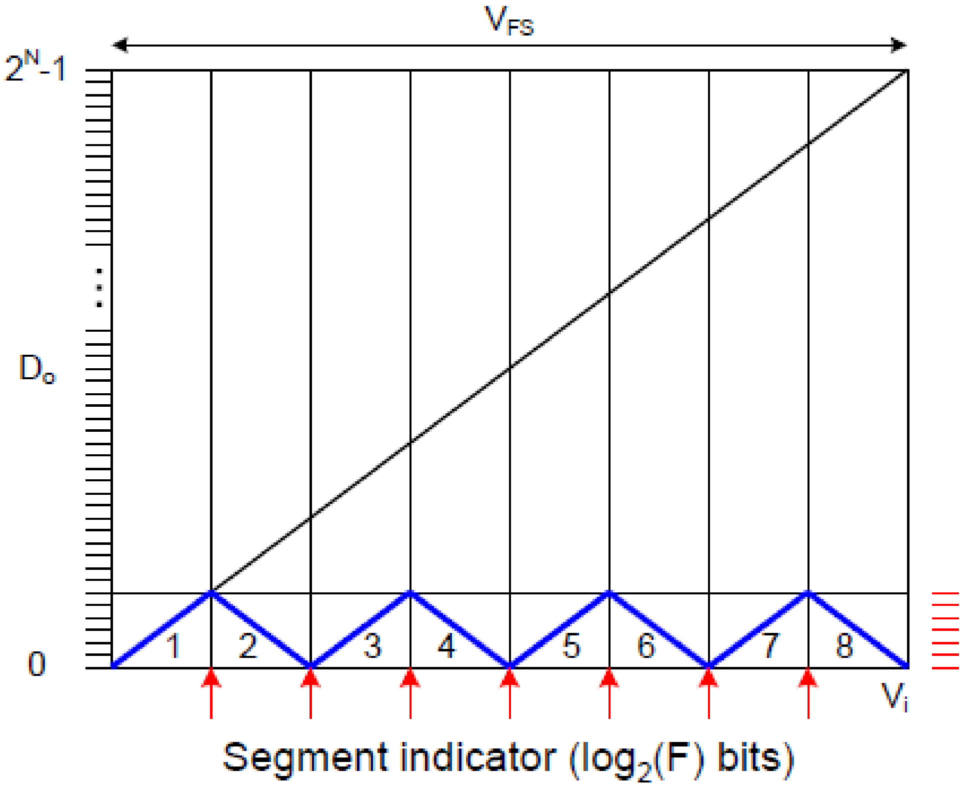
Segmented Quantization



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

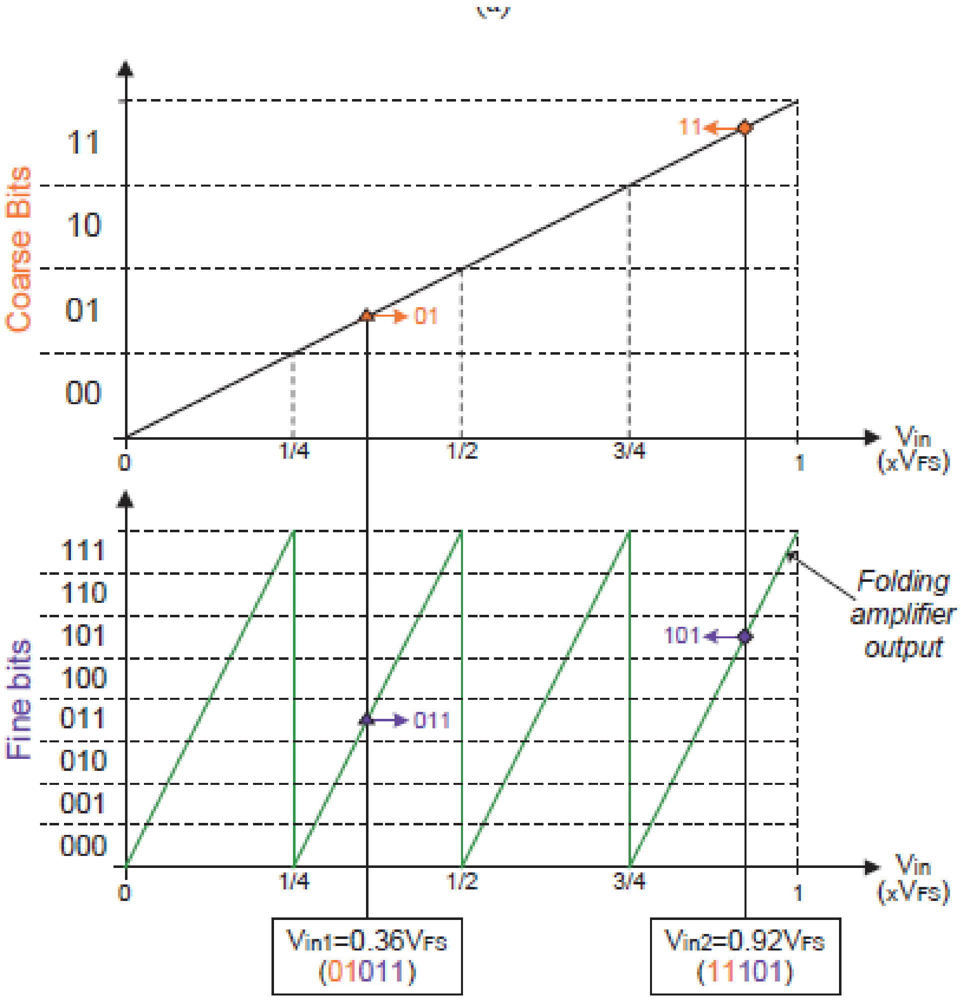
Folding ADCs-Folding Mechanism

Signal Folding



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

Folding ADCs-Folding Mechanism



(b)

A 5-bit example: 2 coarse bits plus 3 fine bits (a) block diagram (b) -generation of coarse and fine bits

Folding ADCs-

Less comparators than Flash ADCs

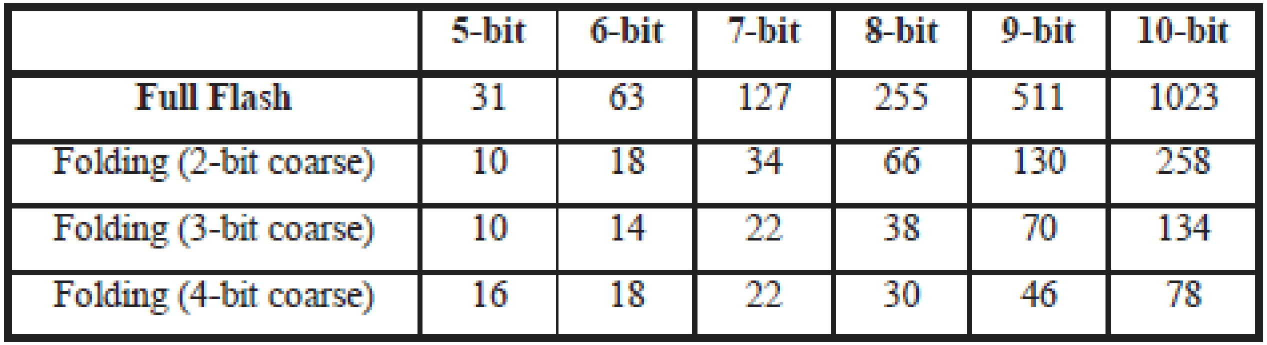
For the 5-bit folding ADC example shown in previous Figure , the whole input range of ADC is divided into four (2exp2) regions, and a 2-bit coarse quantizer can determine one of the four regions where the input voltage falls into. In general, for $F_F = 2^{n_{MSB}}$,

a n_{MSB}*-bit* coarse quantizer is required.

At the same time, the “residue” generated by the folding amplifier is digitized by a 3-bit (n_{LSB}) *fine quantizer.*

Thus, the total number of comparators of this folding A/D converter is 10 (three for the coarse and seven for the fine quantizer), while a 5-bit full-flash ADC need 31 comparators.

Although both Folding ADC and two-step ADC have similar principle, folding ADCs exhibits smaller latency, since In a Folding ADC, fine and coarse information are generated simultaneously,



	5-bit	6-bit	7-bit	8-bit	9-bit	10-bit
Full Flash	31	63	127	255	511	1023
Folding (2-bit coarse)	10	18	34	66	130	258
Folding (3-bit coarse)	10	14	22	38	70	134
Folding (4-bit coarse)	16	18	22	30	46	78

The Table compares the number of comparators in flash and folding ADCs. As the resolution increases, the number of comparators in a folding ADC is much smaller than that of a full flash ADC

Folding ADCs- Folding Signal Implementation

The idea is that simple analog circuits should be used to realize the piece-wise linear input-output characteristics indicated (Figure 27a).
The saw-tooth shaped transfer characteristic is not easy to implement due to its discontinuity. At these discontinued points the slew rate should be infinite, thus a triangular characteristic (Figure 27b) is preferred

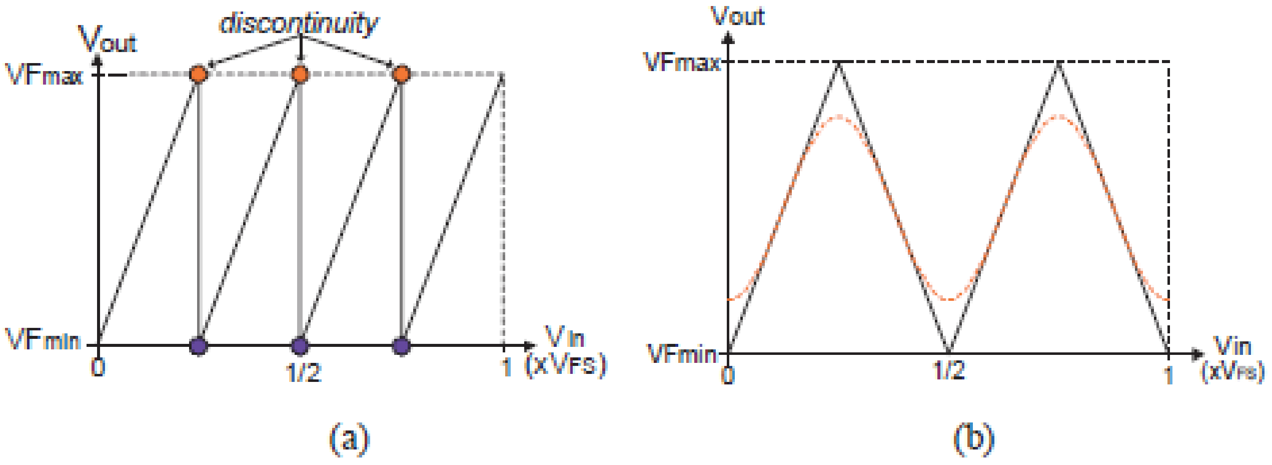


Figure 27 Sawtooth, triangular shaped and pseudo-sinusoidal transfer characteristics (a) sawtooth (b) triangular

Several implementation have been developed which approximate the triangle wave folding characteristic of Figure 27b. Some of them based on rectifier characteristic of Diodes and others based on current mirrors .

Folding ADCs- Folding Signal Implementation

Current-Mirror Based Folding Amplifier

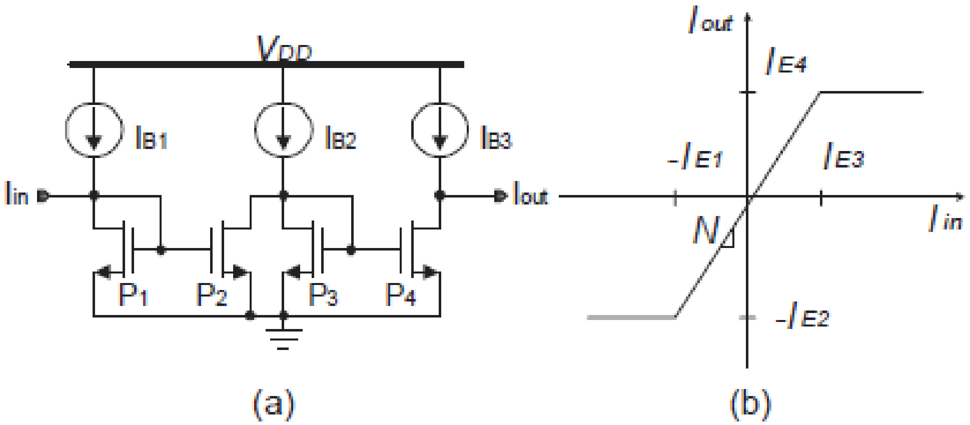


Figure 29 Basic building block of the current mirror based folding amplifier (a) schematic (b) transfer characteristic

Current mirror can be used to implement piecewise linear transfer characteristic of the folding amplifier. The idea is to use basic building block with “S” shaped current-to current transfer characteristics to construct triangular shaped folding waveform.
Schematic of the basic building block is shown in Figure 29a and its transfer curve is shown in Figure 29b.

Folding ADCs- Folding Signal Implementation

Current-Mirror Based Folding Amplifier

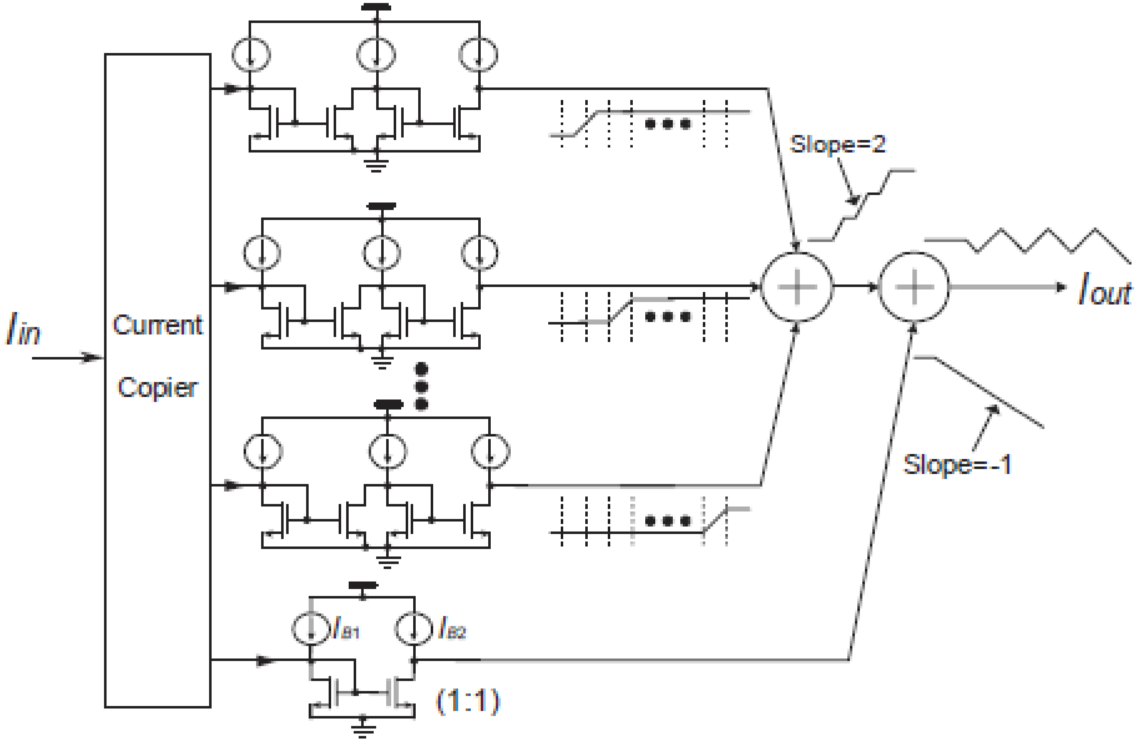


Figure 30 Topology of the current mirror based current mode folding amplifier

Figure 30 shows how the current mode folding amplifier is constructed by connecting basic folding blocks in parallel.

The current copier can be implemented with a PMOS current mirror, which has one input and multiple outputs. By connecting several currents together, a current adder is naturally realized.

Folding ADCs- Folding Signal Implementation

Current-Mirror Based Folding Amplifier

In one word, to implement the nonlinear folding transfer characteristic, current mirrors and transistors comprising them constantly change between “OFF” and “ON” operating states. This will slow down the folding amplifier response.

Generally, circuits with discontinuous input-output characteristics are difficult to realize and are not amenable to high-speed applications. Therefore, folding converters which do not rely upon piece-wise linear folding functions prevail. Folding amplifiers with a “pseudo-sinusoidal” transfer characteristic are much easier to implement than those with a piecewise linear triangular shape transfer characteristic.

Folding ADCs- Folding Signal Implementation

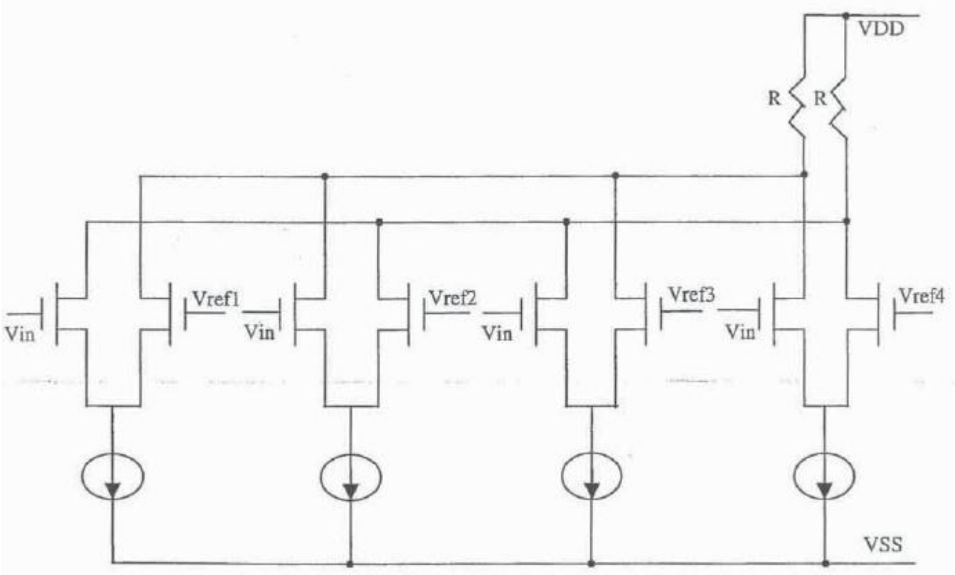
Sinusoidal Folding

Folding amplifiers built with differential pairs have input-output transfer characteristics resembling a sinusoidal signal.

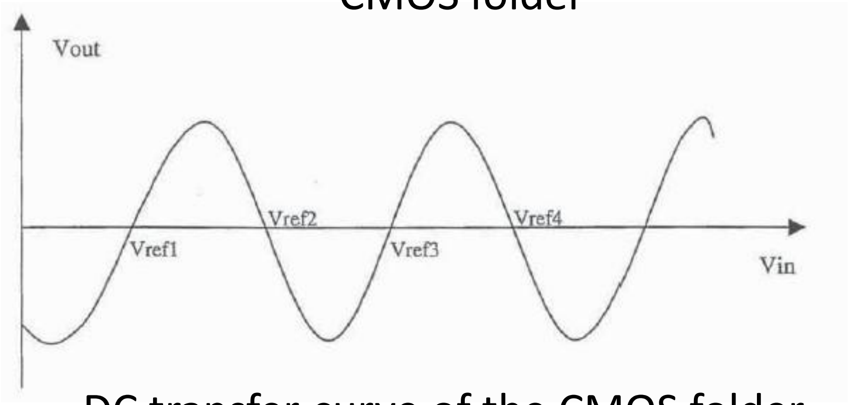
Figure shows the basic scheme for the CMOS folding circuit (a 4-times folder) for use in a 6-bit ADC.

The folder consists of four differential pairs with outputs of the odd and even - number diff-pairs are cross-coupled.

One of the inputs of the diff-pair is connected to the input voltage and the other one is connected to the reference voltage. The outputs of the folder are differential too.



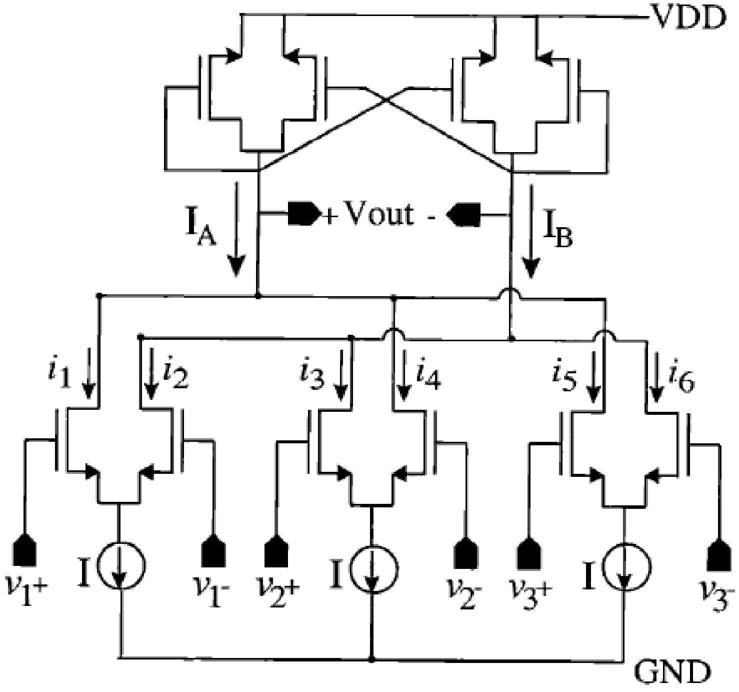
CMOS folder



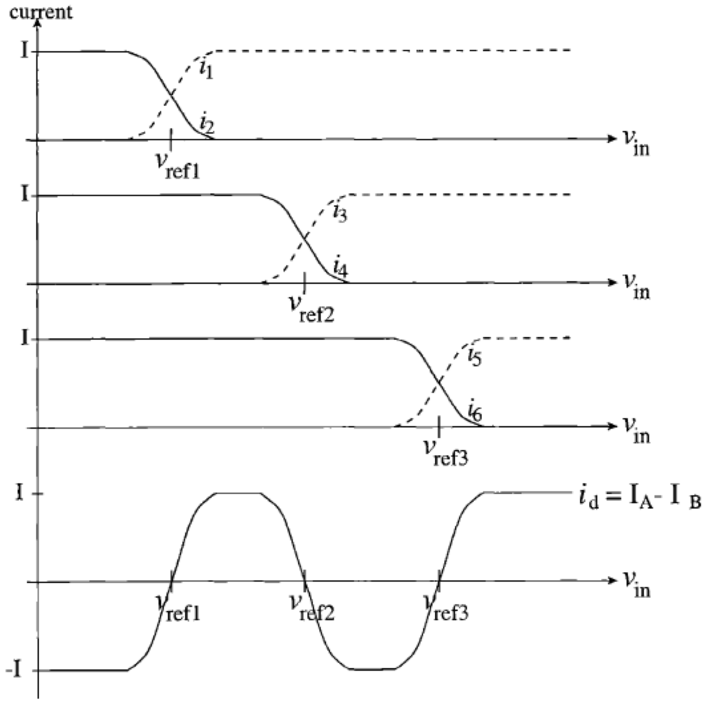
DC transfer curve of the CMOS folder

Folding ADCs- Folding Signal Implementation

Graphical Explanation of sinusoidal folding signal creation from diff-amplifiers



Folding amplifier circuit with a folding factor of three.



Signal waveforms of the 3-fold differential amplifier

Folding ADCs- Double Folding

Nevertheless, the perfectly linear triangular curve is difficult to generate, and its corners tend to become rounded. In order to compensate for this problem, two folding circuits are used in parallel to generate two folding signals that have a carefully calculated mutual offset. This will guarantee that at least one signal is operating in a reasonably linear region for all inputs. This idea is illustrated in Figure 2-3 below. Each folding signal only has to stay linear within the region which it is in use. It is also important to notice that when two signals are used, the range of v_{out} to be detected is further reduced by a factor of two. This demonstrates that the number of voltage levels that need to be distinguished per folding signal can be interchanged to the number of folding signals used.

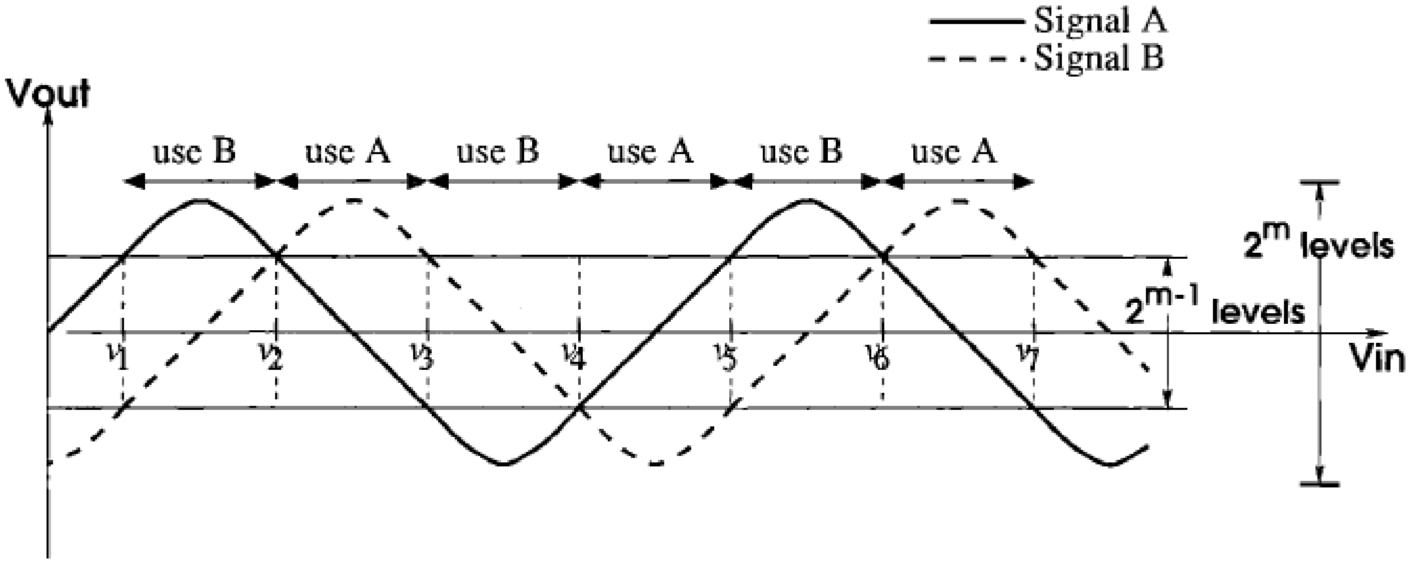


Figure 2-3: The use of a second folding signal for overcoming nonlinearity in corner region.

Folding ADCs- Double Folding

Here, in a single folding system (upper part), the full scale input of the ADC is divided into 4 segments (1-4) and each segment corresponds to full range of the 3-bit quantizer, thus a strict piece-wise linear transfer characteristic is desired

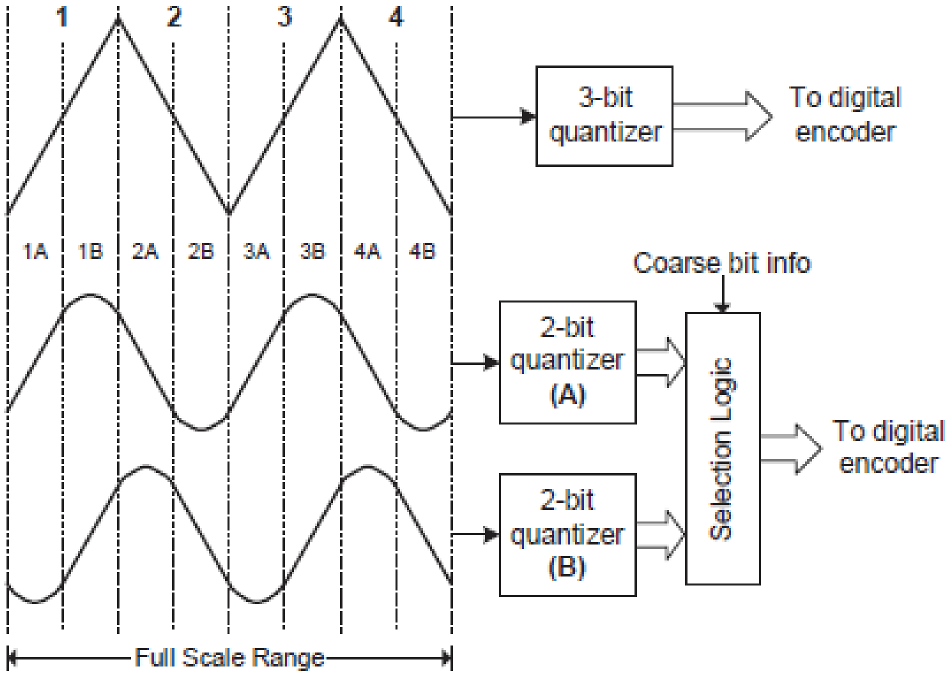


Figure 33 Comparison of single and double folding system

In a double folding system, the ADC full input range is divided into 8 segments, each of the 2 quantizers handle 4 segments, i.e. quantizer (A) digitize 1A-4A while segments 1B-4B belong to quantizer (B). The selection logic block always chooses the output of the quantizer the folding amplifier of which is in linear region. If one folding signal is in its nonlinear region, the other is in its linear region and vice versa.

Thus, instead of needing one good folding signal with the detection of 8 levels, which a 3-bit quantizer demands, we also can take two folding signals with the detection of 4 levels for each folding signal.

Folding ADCs- Multiple Folding

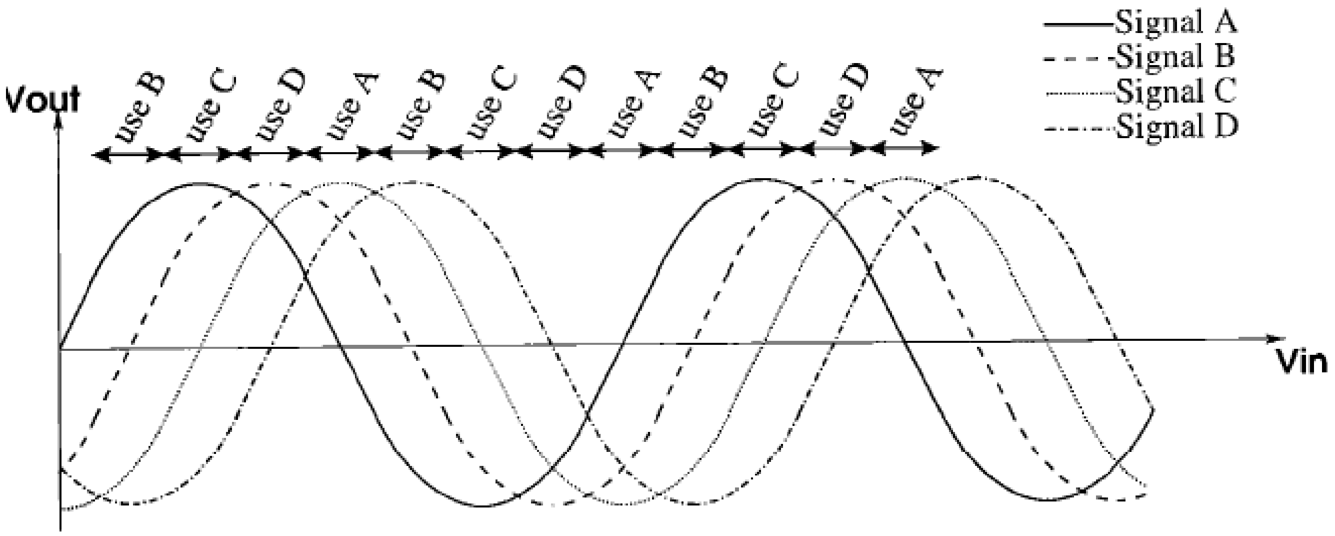
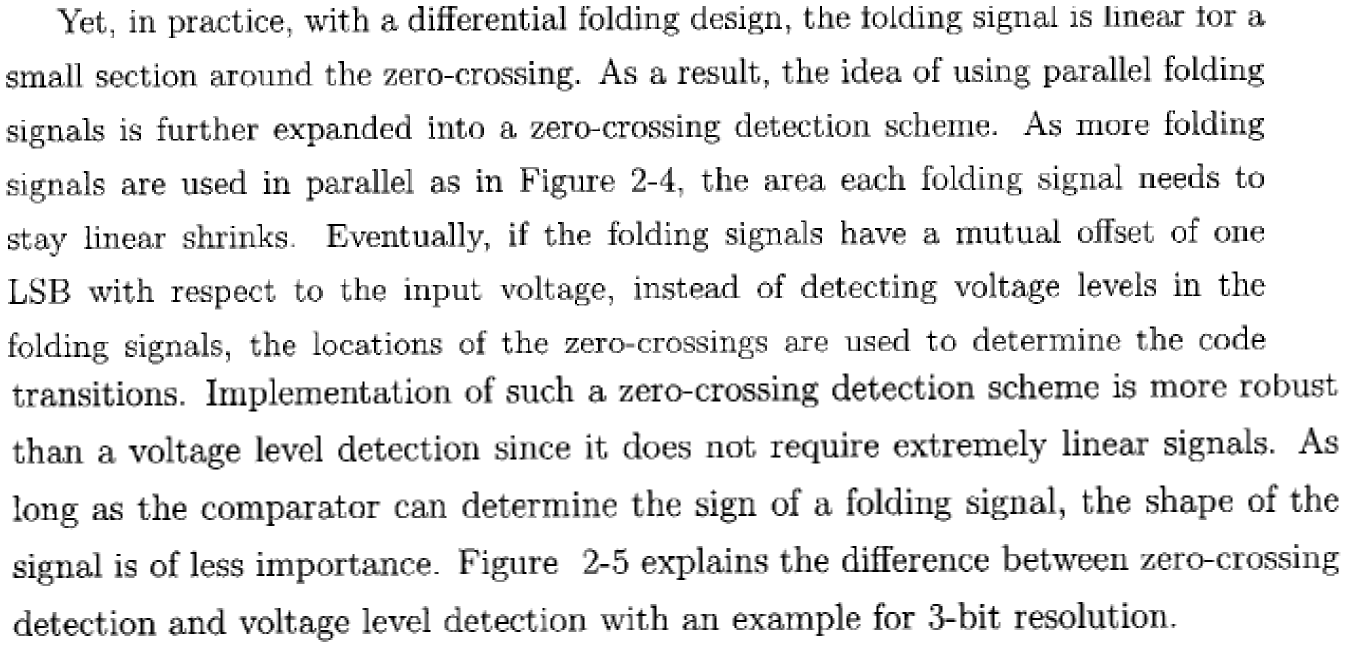
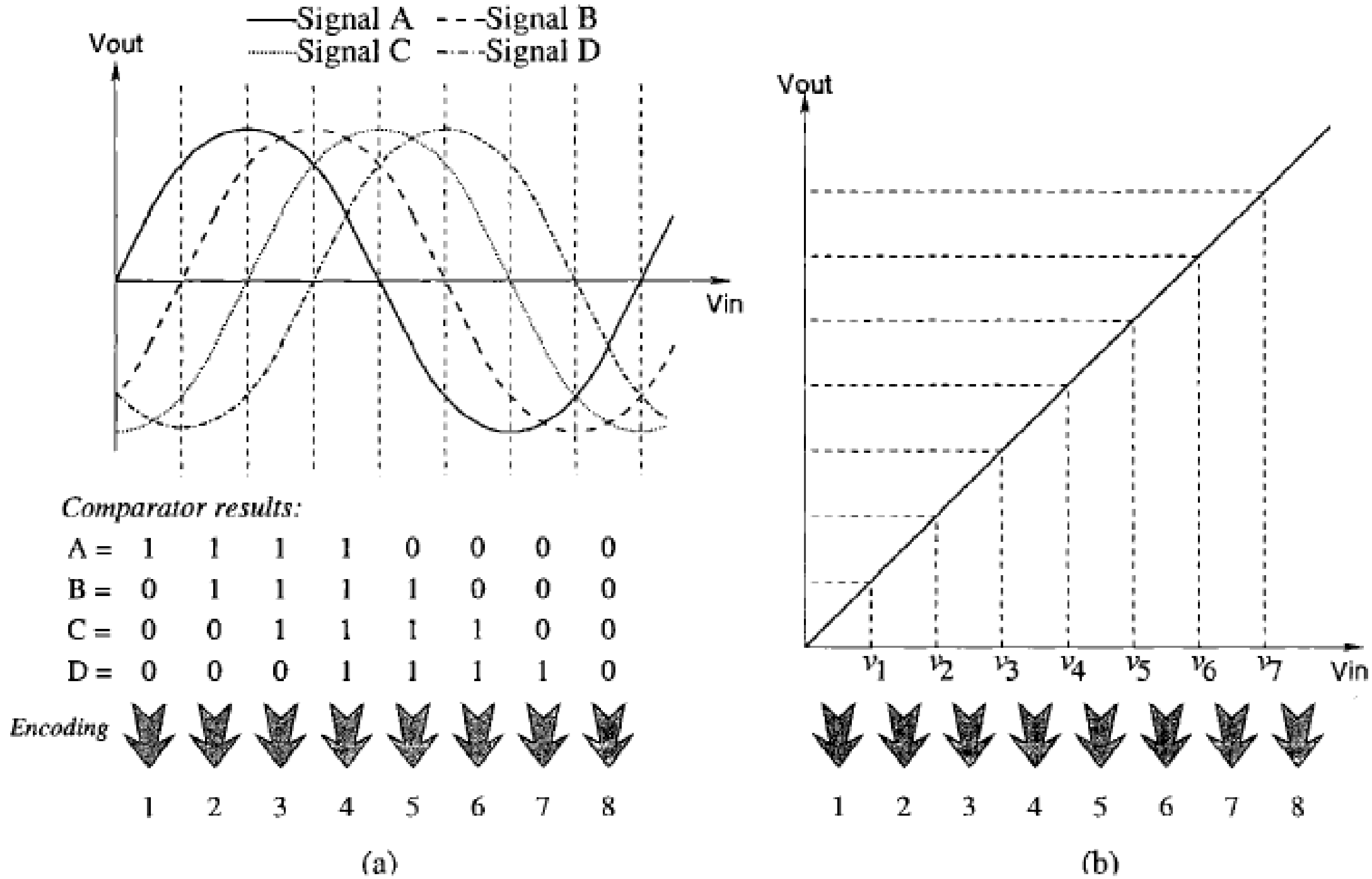


Figure 2-4: The use of four folding signals for further shrinking the linear region required for each folding signal.



Yet, in practice, with a differential folding design, the folding signal is linear for a small section around the zero-crossing. As a result, the idea of using parallel folding signals is further expanded into a zero-crossing detection scheme. As more folding signals are used in parallel as in Figure 2-4, the area each folding signal needs to stay linear shrinks. Eventually, if the folding signals have a mutual offset of one LSB with respect to the input voltage, instead of detecting voltage levels in the folding signals, the locations of the zero-crossings are used to determine the code transitions. Implementation of such a zero-crossing detection scheme is more robust than a voltage level detection since it does not require extremely linear signals. As long as the comparator can determine the sign of a folding signal, the shape of the signal is of less importance. Figure 2-5 explains the difference between zero-crossing detection and voltage level detection with an example for 3-bit resolution.

Folding ADCs- Multiple Folding



Comparator results:

A =	1	1	1	1	0	0	0	0
B =	0	1	1	1	1	0	0	0
C =	0	0	1	1	1	1	0	0
D =	0	0	0	1	1	1	1	0

Figure 2-5: (a) A zero-crossing detection scheme where the locations of zero-crossings are used to determine the code transitions. This alleviates the problem of requiring a perfectly linear signals. (b) A voltage level detection scheme where the linear signal is compared to reference voltages (v_1 to v_7) to determine code transitions.

Folding ADCs- Multiple Folding

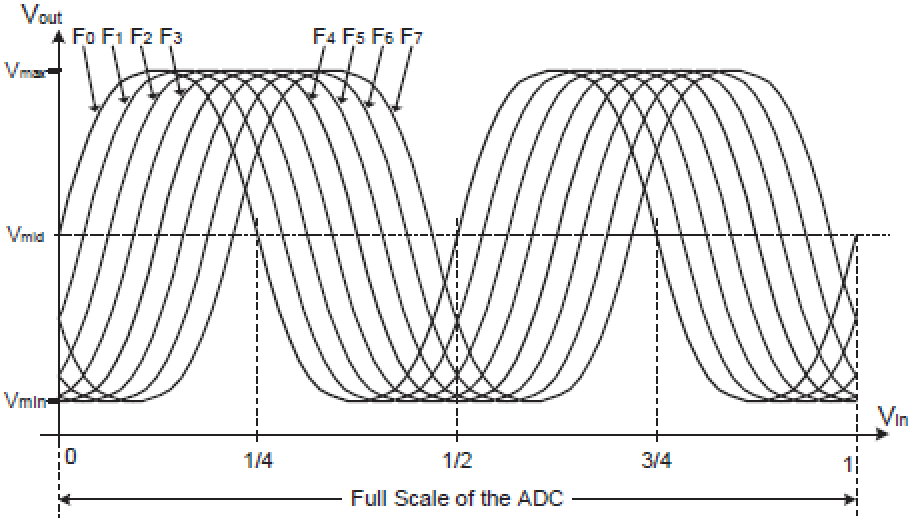


Figure 34 8 Folding waveforms generate all 32 zero crossings of a 5 bit ADC

Figure 34 shows all the 8 waveforms of an 8-folding system. These eight folding waveforms generate 32 (5-bit) equidistant zero-crossing points along the full ADC input range. Thus, linearity of each folding waveform is no longer critical, only the positions of zero crossing points are of interest, which affect the linearity of the folding ADC. For the folding ADC example shown in Figure 34, the number of zero crossing detection comparators is 8(fine quantizer) plus 3(coarse quantizer). A 5-bit full-flash will need 31 comparators. The problem is now that the generation of 8 folding signals with 8*5 differential pairs is as much hardware as a full-flash converter. Interpolation can be used to circumvent this dilemma.

Folding ADCs- Interpolation

As more folding signals are used, it starts to impose limitations because each folding signal requires a different folding amplifier. Thus, the associated hardwares will increase, and the complexity and power consumption will be comparable to a flash converter. Therefore, the technique of interpolation is introduced.

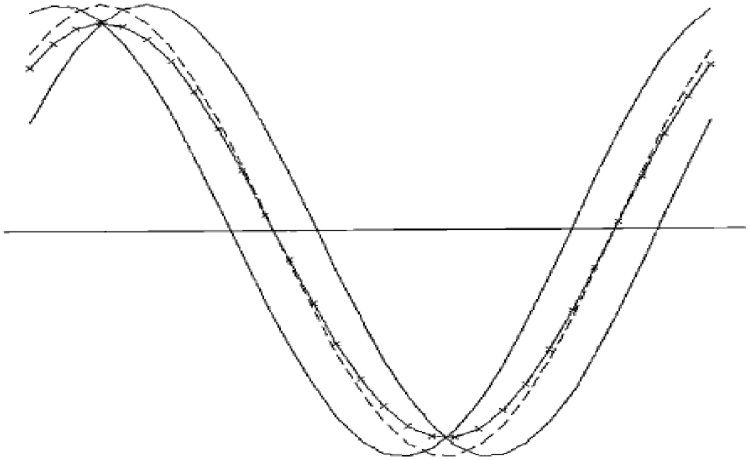
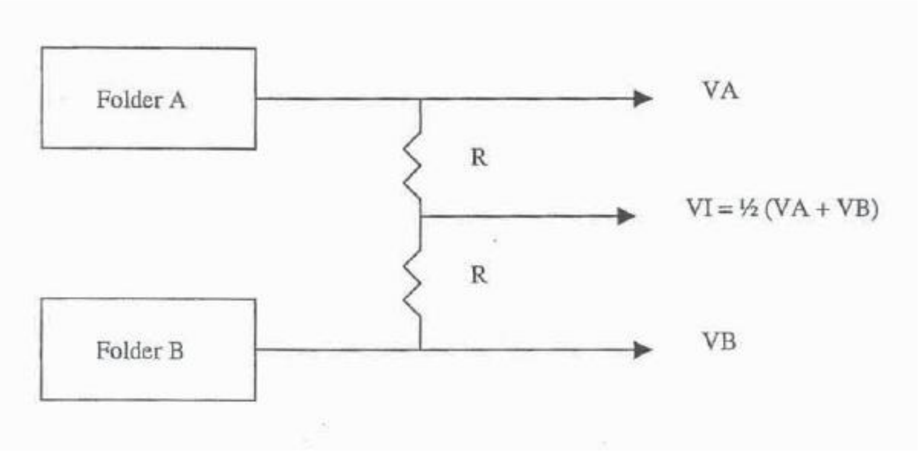
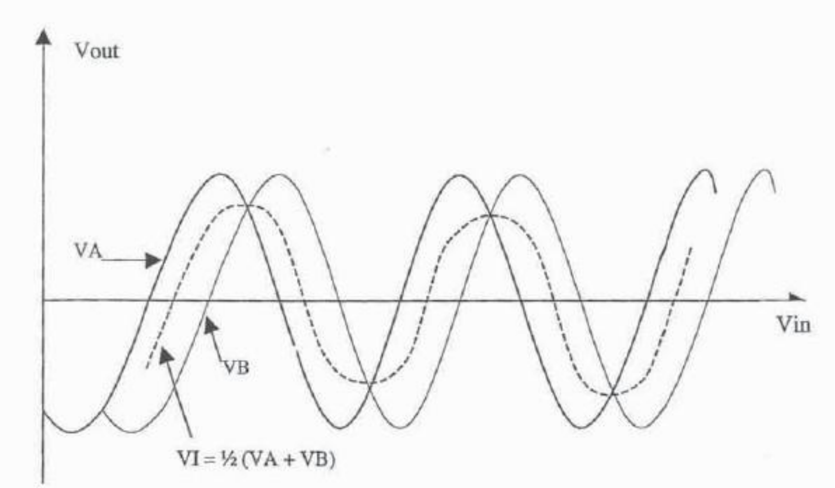


Figure 2-6: Approximation of the third folding signal by interpolation: The two solid lines are the two existing folding signals. The dashed line is the third ideal folding signal. The solid line with crosses is an approximation of the third folding signal by interpolating between the two folding signals.

As demonstrated in Figure 2-6, the dotted curve is the desired folding signal, and the solid line with crosses is an approximation of it obtained by interpolating between the other two solid curves. Although the interpolated and ideal signals have different amplitude, they have the same zero-crossing locations; hence, they give rise to the same result when employed in the zero-crossing detection scheme described in the last section. Since interpolation can be easily implemented by a resistors divider, this approach will save hardware.

Folding ADCs- Interpolation





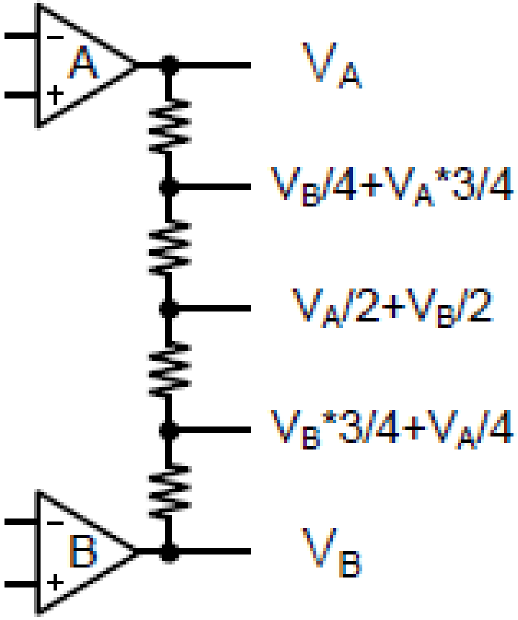
The basic principle of interpolation is shown in Figure 2.2.1 and 2.2.2. Folder A and B generate two shifted folding signals, V A and VB. Another folding signal that lies between V A and VB can be generated using the resistor chain (averaging).

This being the case, for a 6-bil folding ADCs, we only need 8 folders and can use the interpolating circuit to generate the other 8 folding signals.

Note that the top and bottom of the interpolated signals are somewhat non ideal. This is not important, however, since only the zero-crossing points are actually used.

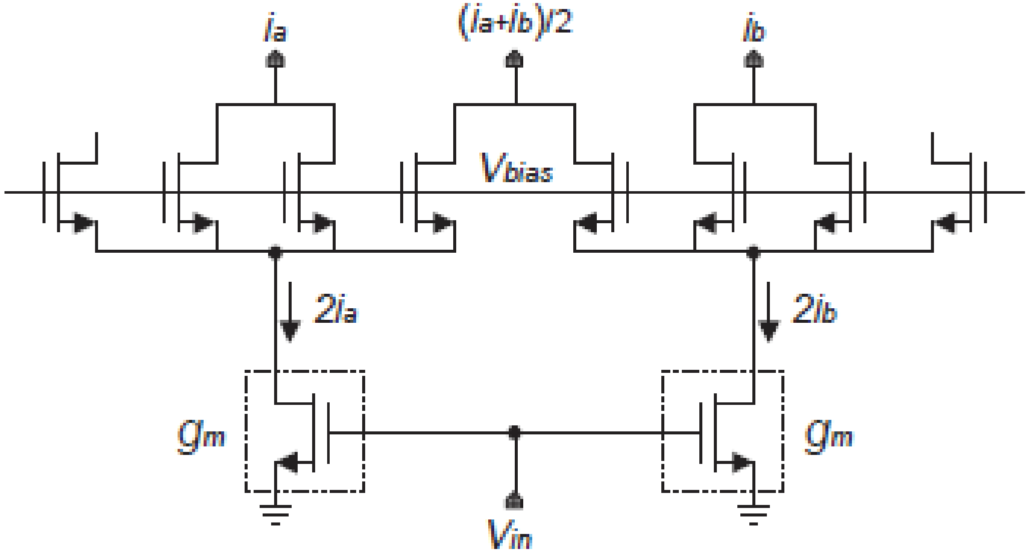
Folding ADCs- Interpolation

Interpolation factor of 4



Folding ADCs- Current Interpolation

The interpolating currents are split with cascode current mirrors into various fractions proportional to the current mirror size and are summed to form the fine current divisions



Current mode interpolation based on current splitting

Less accurate then voltage interpolation due to mismatch of current mirrors

Folding ADCs- Total Architecture

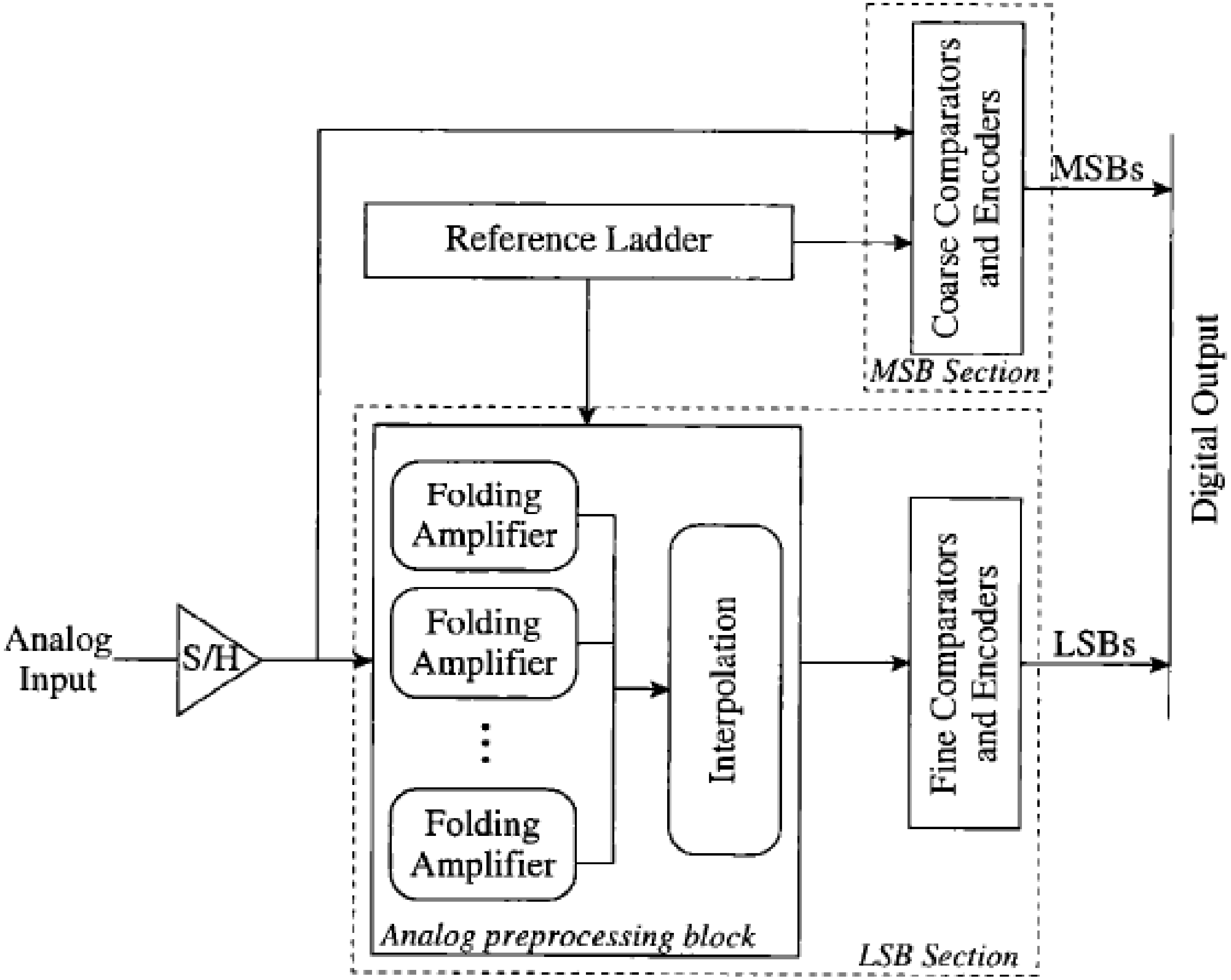


Figure 3-1: A classical folding and interpolating ADC.

Folding ADCs- LSBs’ Architecture

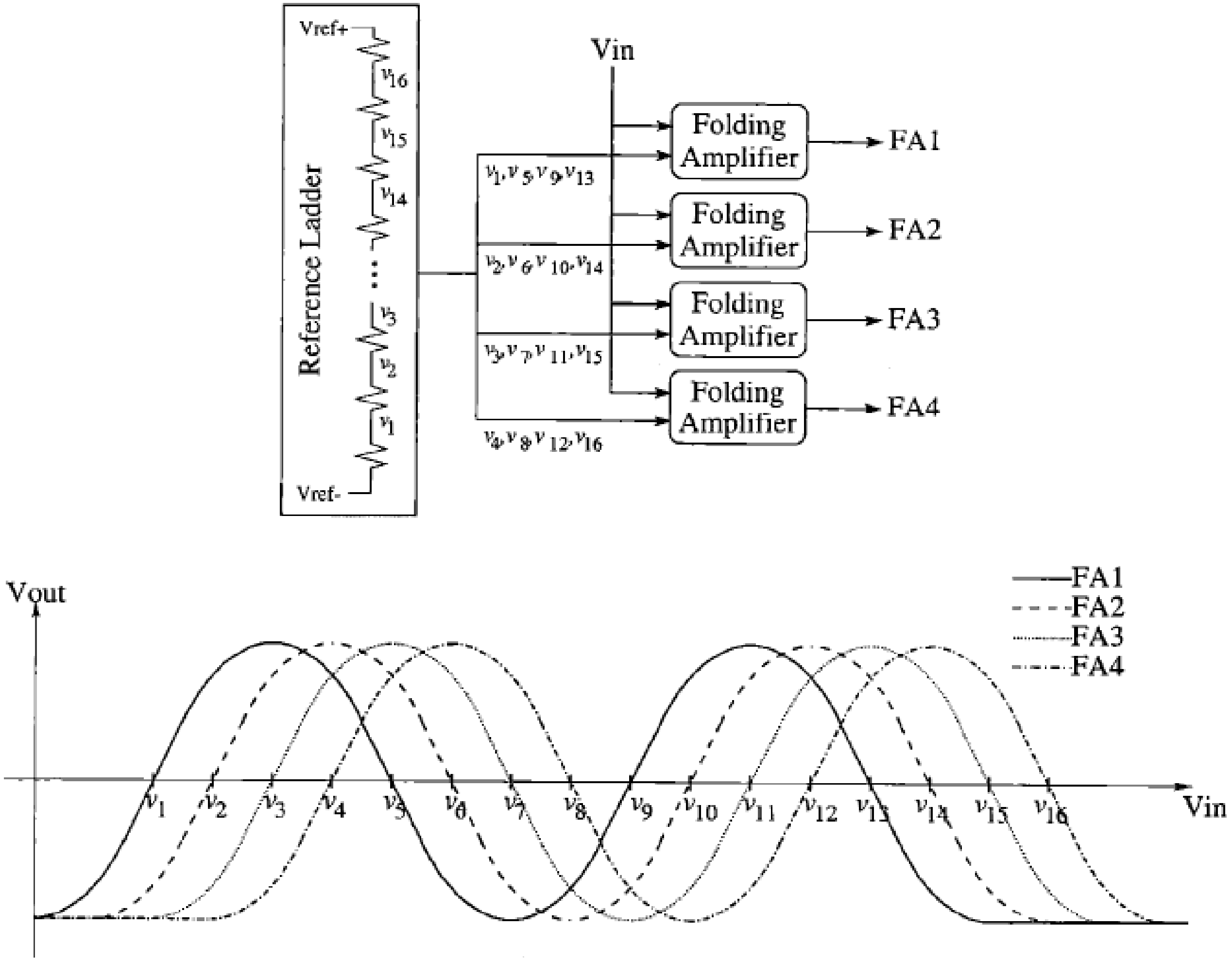

Reference Ladder
Vref+
Vref-
v_{16}
v_{15}
v_{14}
...
v_3
v_2
v_1
Vin
v_1, v_5, v_9, v_{13}
v_2, v_6, v_{10}, v_{14}
v_3, v_7, v_{11}, v_{15}
v_4, v_8, v_{12}, v_{16}
Folding Amplifier
Folding Amplifier
Folding Amplifier
Folding Amplifier
FA1
FA2
FA3
FA4
Vout
Vin
v_1 v_2 v_3 v_4 v_5 v_6 v_7 v_8 v_9 v_{10} v_{11} v_{12} v_{13} v_{14} v_{15} v_{16}
FA1
FA2
FA3
FA4

Figure 3-4: An implementation of four parallel folding amplifiers to create four folding signals which are 45 degrees out of phase with each other.

Folding ADCs- LSBs' Architecture

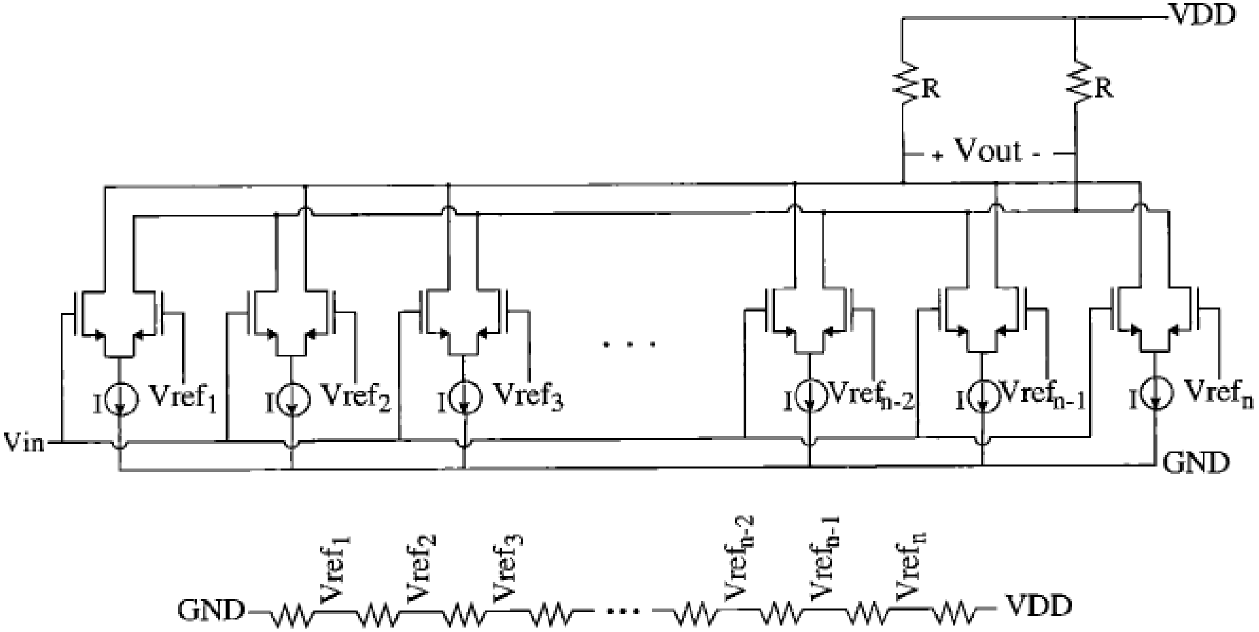


Figure 3-2: A typical implementation of a folding amplifier.

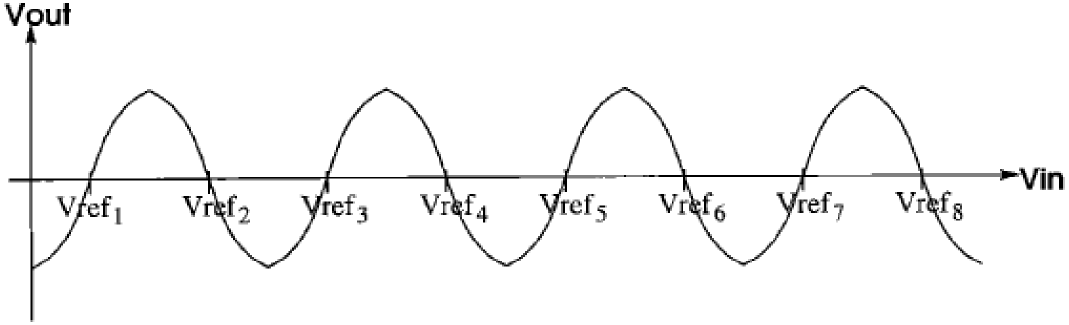
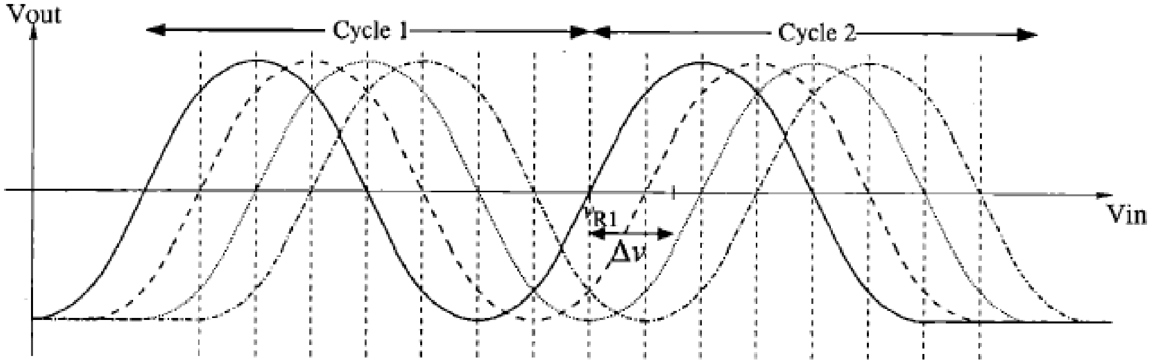


Figure 3-3: A folding signal created by a folding amplifier with a folding factor of eight.

Folding ADCs- MSB Section

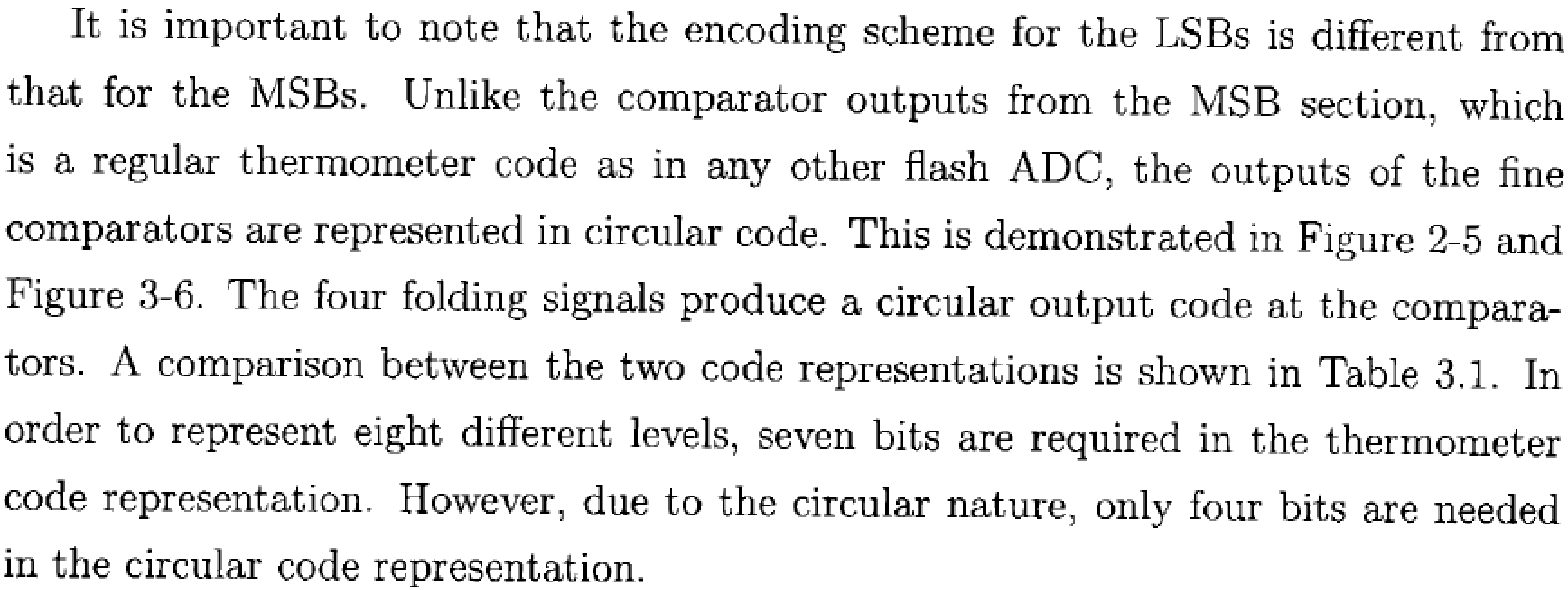


	input section	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16
L S B	comp output	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
		0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
		0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0
		1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
											Encoding						
	binary output	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111
M S B	comp output	Subrange 1 0								Subrange 2 1							
Final Digital Output		0	0	0	0	0	0	0	0	0	0 1	1	1	1	1	1	1
		0	0	0	0	1	1	1	1	0	0 0	0	0	1	1	1	1
		0	0	1	1	0	0	1	1	0	0 0	1	1	0	0	1	1
		0	1	0	1	0	1	0	1	0	1 1	0	1	0	1	0	1

error

The MSB section is essentially a low-resolution flash ADC. It comprises of a group of coarse comparators which can quickly locate in which cycle the input signal lies after the folding process, thus determining the MSBs' values. It is illustrated with an example in Figure 3-6. The input signal is compared with a reference voltage set at v_{R1}. The comparator outputs a zero if the input lies between zero and v_{R1}. If the input is greater than v_{R1}, the comparator output is set to one. This result can then be used to set the MSB.

Encoding Logic (for ouputing final binary code)



It is important to note that the encoding scheme for the LSBs is different from that for the MSBs. Unlike the comparator outputs from the MSB section, which is a regular thermometer code as in any other flash ADC, the outputs of the fine comparators are represented in circular code. This is demonstrated in Figure 2-5 and Figure 3-6. The four folding signals produce a circular output code at the comparators. A comparison between the two code representations is shown in Table 3.1. In order to represent eight different levels, seven bits are required in the thermometer code representation. However, due to the circular nature, only four bits are needed in the circular code representation.

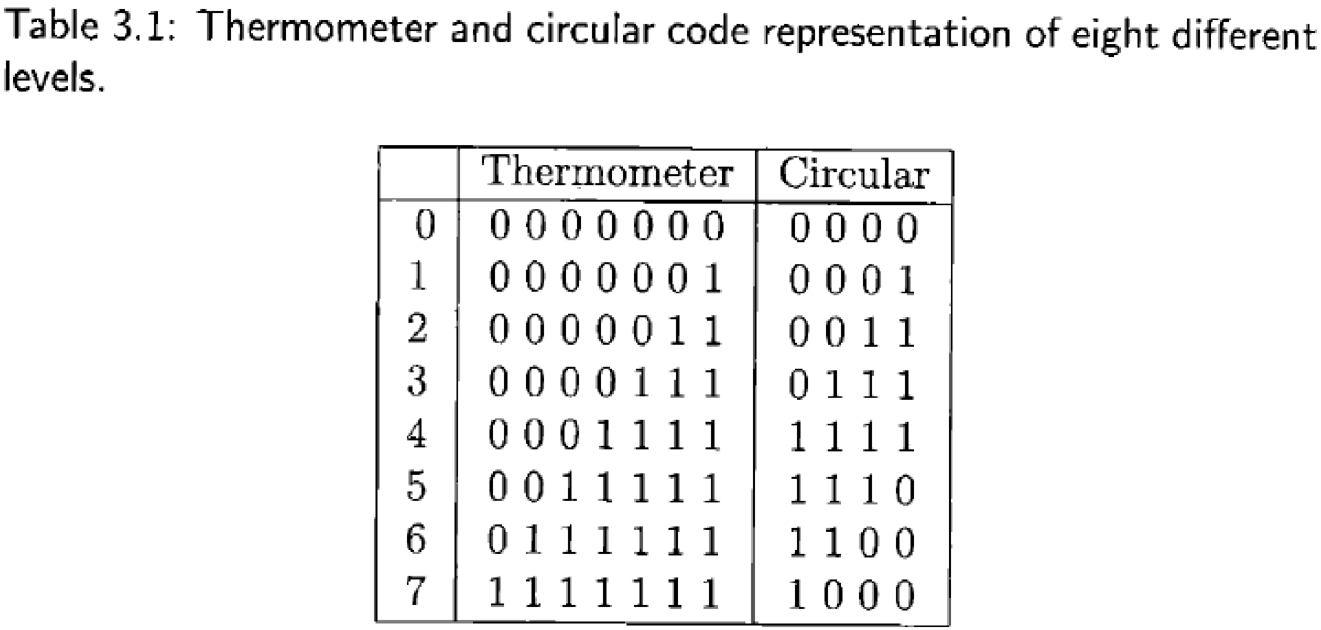


Table 3.1: Thermometer and circular code representation of eight different levels.

	Thermometer	Circular
0	0 0 0 0 0 0 0	0 0 0 0
1	0 0 0 0 0 0 1	0 0 0 1
2	0 0 0 0 0 1 1	0 0 1 1
3	0 0 0 0 1 1 1	0 1 1 1
4	0 0 0 1 1 1 1	1 1 1 1
5	0 0 1 1 1 1 1	1 1 1 0
6	0 1 1 1 1 1 1	1 1 0 0
7	1 1 1 1 1 1 1	1 0 0 0

Sigma-Delta A/DConverters

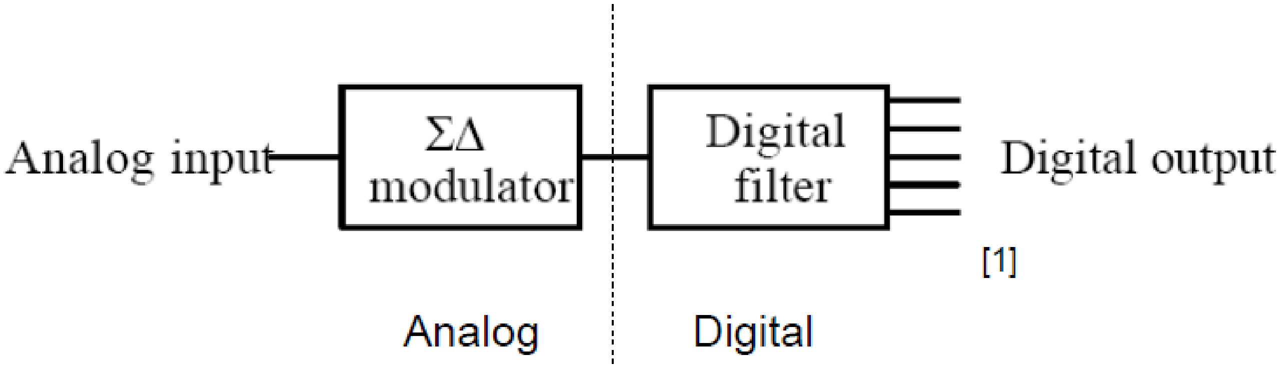
The IC Σ-Δ ADC offers several advantages over the other architectures, especially for high resolution, low frequency applications. First and foremost, the single-bit Σ-Δ ADC is inherently monotonic. The Σ-Δ ADC also lends itself to low cost foundry CMOS processes because of the digitally intensive nature of the architecture.

Modern CMOS Σ-Δ ADCs (and DACs, for that matter) are the converters of choice for voiceband and audio applications. The highly digital architectures lend themselves nicely to fine-line CMOS. In addition, high resolution (up to 24 bits) low frequency Σ-Δ ADCs have virtually replaced the older integrating converters in precision industrial measurement applications

Sigma-Delta A/DConverters

BASICS OF Σ-Δ ADCS

A Σ-Δ ADC contains very simple analog electronics –the ΣΔ modulator:
(a comparator, voltage reference, a switch, and one or more integrators and analog summing circuits),
and quite complex digital computational circuitry. This digital circuitry consists of a digital signal processor (DSP) which acts as a filter (generally, a low pass filter).



To understand how a Σ-Δ ADC works, familiarity with the concepts of *oversampling, quantization noise shaping, digital filtering, and decimation is required.*

Sigma-Delta A/DConverters

BASICS OF Σ-Δ ADCS

The following are brief definitions of terms that will be used henceforth:

Noise Shaping Filter or Integrator: The noise shaping filter or integrator of a sigma delta converter distributes the converter quantization error or noise such that it is very low in the band of interest.

Oversampling. Oversampling is simply the act of sampling the input signal at a frequency much greater than the Nyquist frequency (two times the input signal bandwidth). Oversampling decreases the quantization noise in the band of interest.

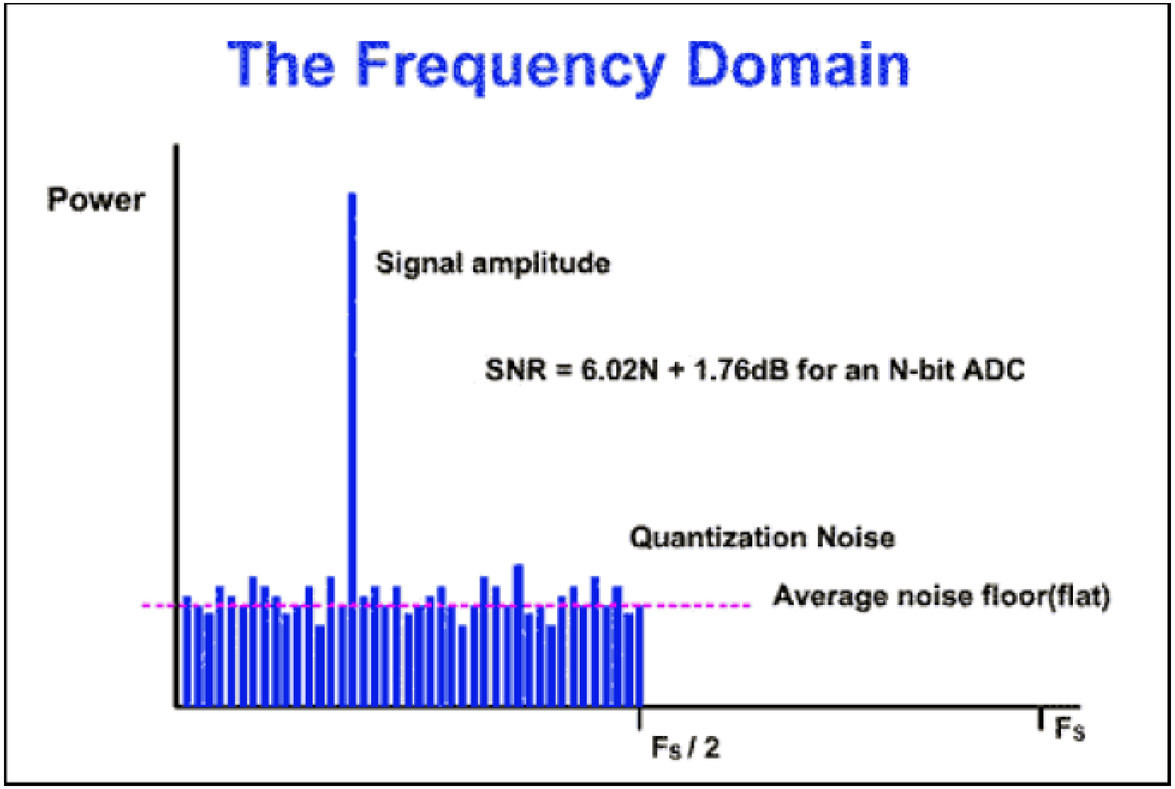
Digital Filter. An on-chip digital filter is used to attenuate signals and noise that are outside the band of interest.

Decimation: Decimation is the act of reducing the data rate down from the oversampling rate without losing information.

Sigma-Delta A/DConverters

Oversampling

First, consider the frequency-domain transfer function of a traditional multi-bit ADC with a sine-wave input signal. This input is sampled at a frequency Fs. According to Nyquist theory, Fs must be at least twice the bandwidth of the input signal



If we divide the fundamental amplitude by the RMS sum of all the frequencies representing noise, we obtain the signal to noise ratio (SNR).

For an N-bit ADC, SNR = 6.02N + 1.76dB. To improve the SNR in a conventional ADC (and consequently the accuracy of signal reproduction) you must increase the number of bits.

Sigma-Delta A/DConverters

Oversampling

Consider again the above example, but with a sampling frequency increased by the oversampling ratio k, to kFs

An FFT analysis shows that the noise floor has dropped. SNR is the same as before, but the noise energy has been spread over a wider frequency range. Sigma-delta converters exploit this effect by following the 1-bit ADC with a digital filter **The RMS noise is less, because most of the noise passes through the digital filter. This** action enables sigma-delta converters to achieve wide dynamic range from a low-resolution ADC.

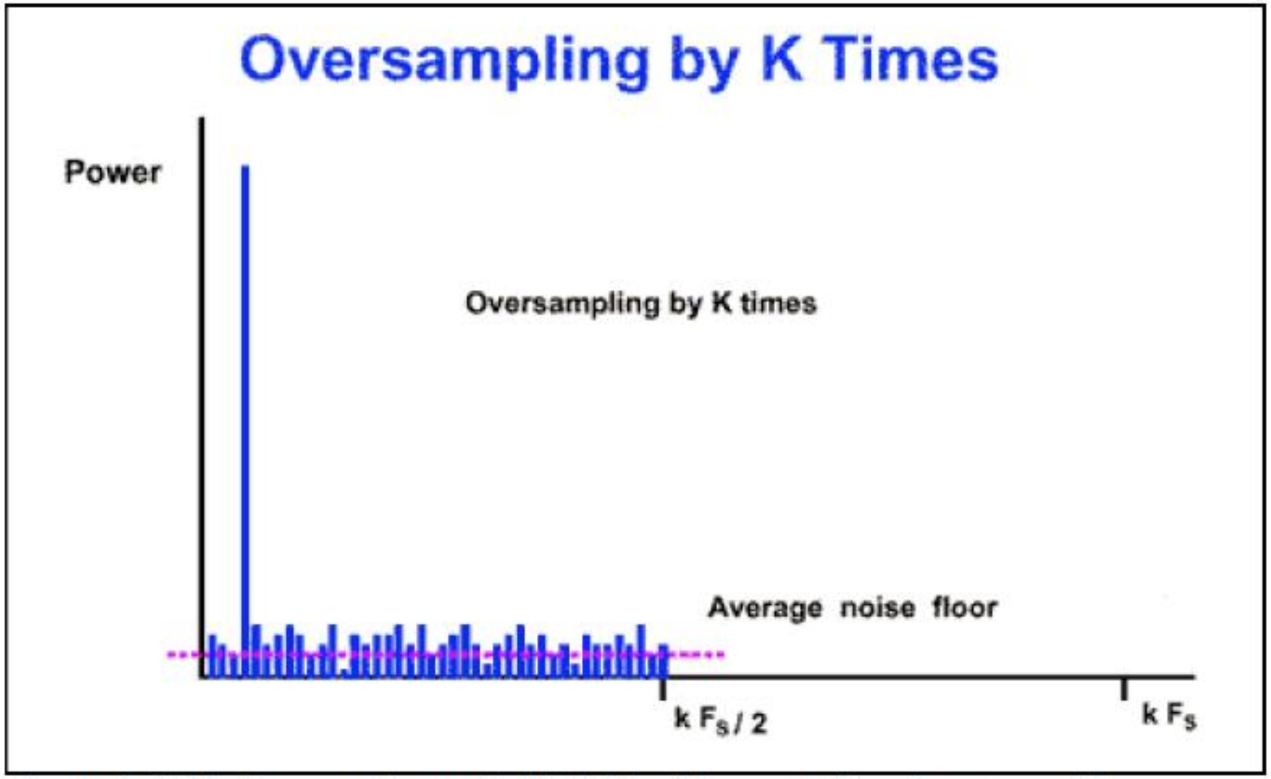


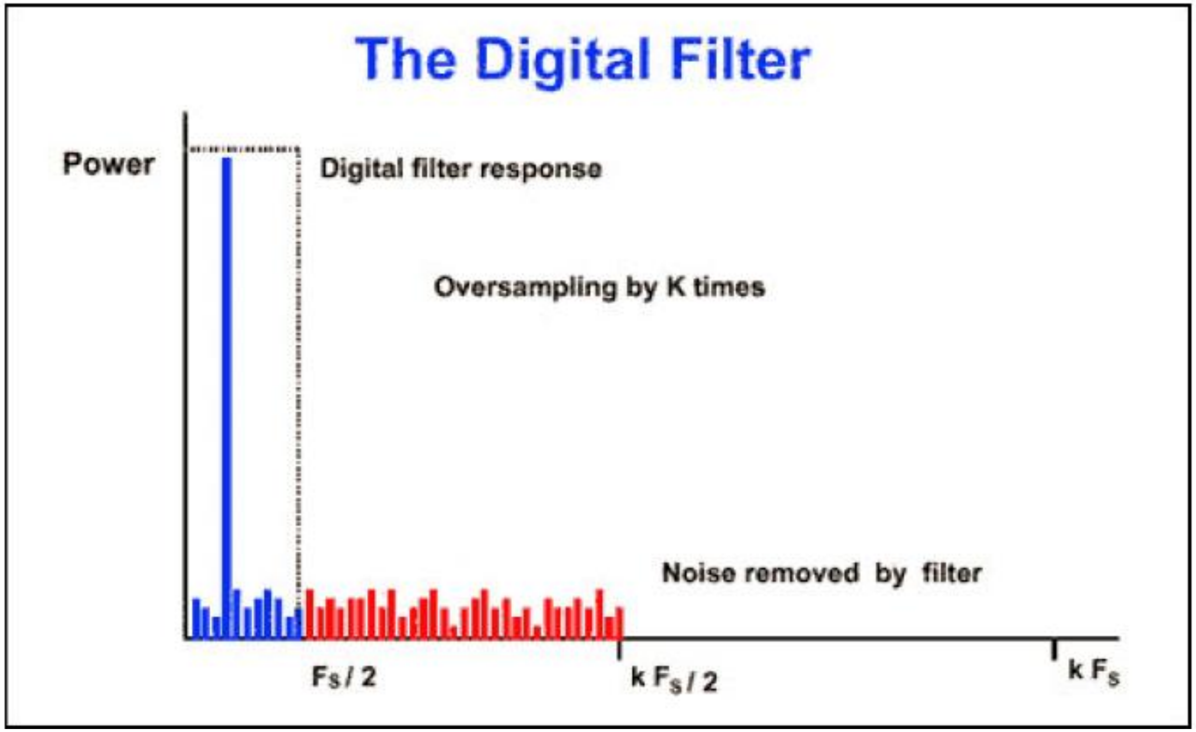
Figure 2. FFT diagram of a multi-bit ADC with a sampling frequency kF_S.

Sigma-Delta A/DConverters

Oversampling+Digital Filtering *Effect of the digital filter on the noise bandwidth*

$$n_0 = e_{RMS}\left(\frac{2f_0}{f_S}\right)^{\frac{1}{2}}$$

n0 = in-band quantization noise, f0 is the input signal bandwidth, and fs = sampling frequency. The quantity fs/2f0 is generally referred to as the Oversampling Ratio or OSR. Equation above shows that oversampling reduces the in band quantization noise by the square root of the OSR



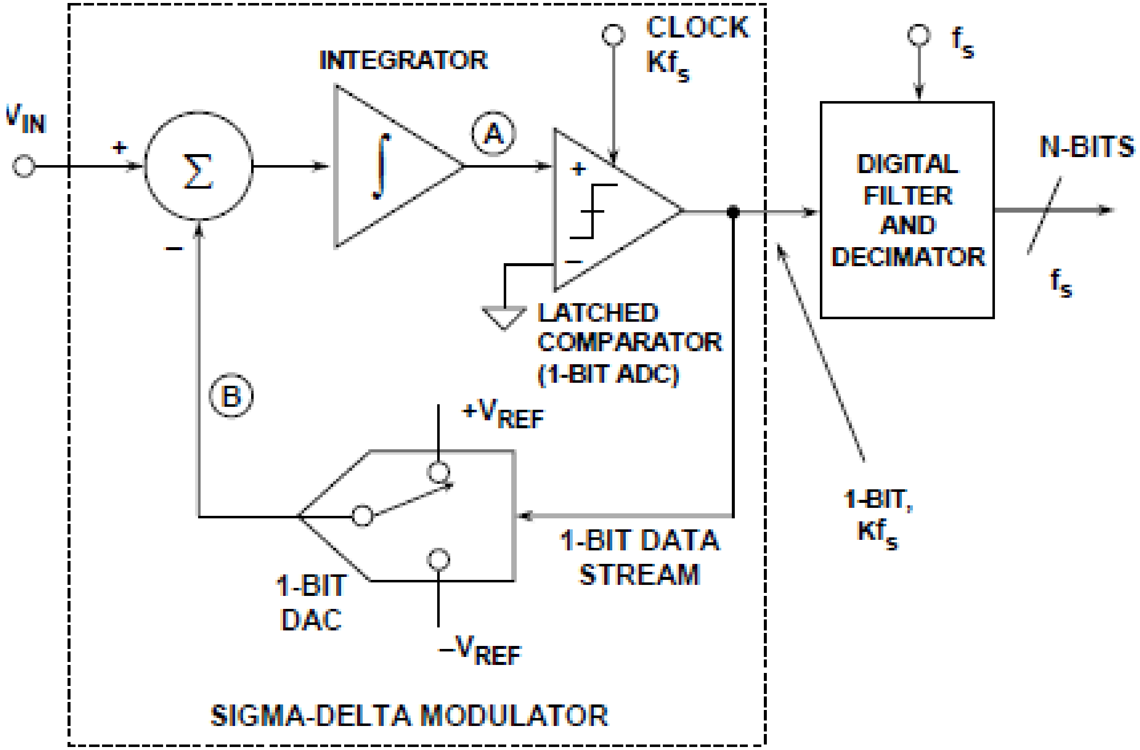
Does the SNR improvement come simply from oversampling and filtering? Note that the SNR for a 1-bit ADC is 7.78dB (6.02 + 1.76). Each factor-of-4 oversampling increases the SNR by 6dB, and each 6dB increase is equivalent to gaining one bit.

A 1-bit ADC with 24x oversampling achieves a resolution of four bits, and to achieve 16-bit resolution you must oversample be a factor of 4exp15, which is not realizable. But, sigma-delta converters overcome this limitation with the technique of noise shaping, which enables a gain of more than 6dB for each factor of 4x oversampling.

Sigma-Delta A/DConverters

Noise Shaping-Explanation of Σ-Δ mechanism

To understand noise shaping, consider the block diagram of a sigma-delta modulator of the first order**. It** includes a difference amplifier, an integrator, and a comparator with feedback loop that contains a 1-bit DAC. Intuitively, a Σ-Δ ADC operates as follows. Assume a dc input at VIN. The integrator is constantly ramping up or down at node A. The output of the comparator is fed back through a 1-bit DAC to the summing input at node B. The negative feedback loop from the comparator output through the 1-bit DAC back to the summing point will force the average dc voltage at node B to be equal to VIN. This implies that the average DAC output voltage must equal the input voltage VIN. The average DAC output voltage is controlled by the *ones-density in the 1-bit data stream from the comparator output.*



Sigma-Delta A/DConverters

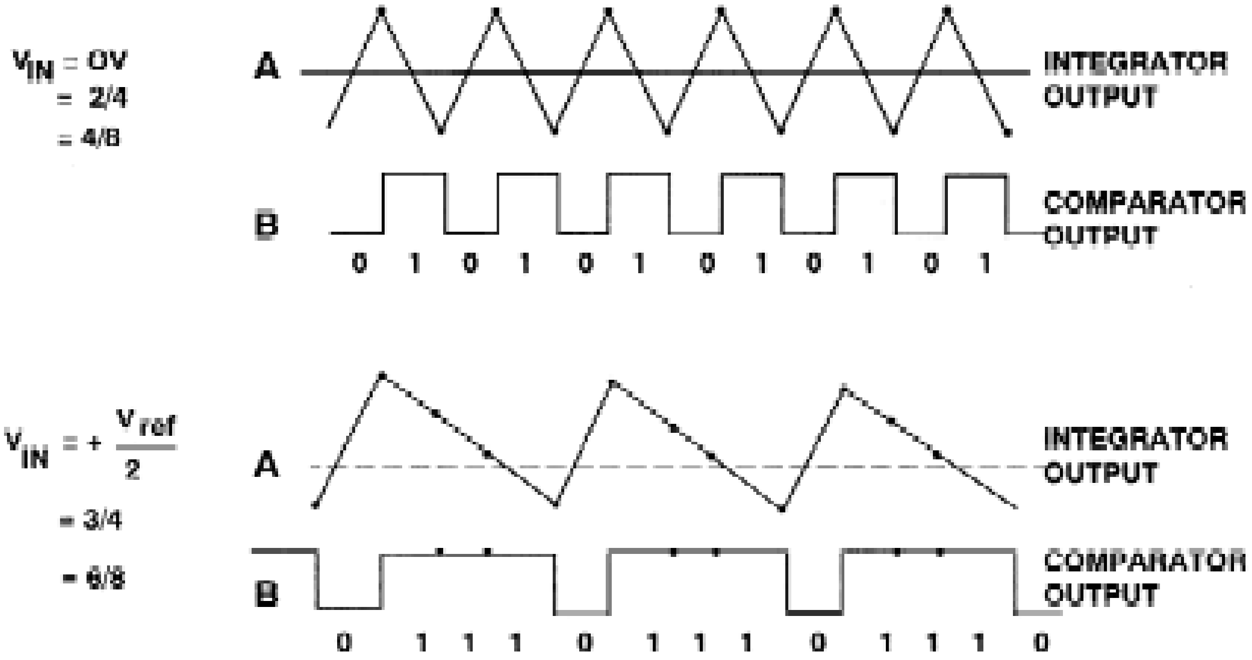
Noise Shaping-Explanation of Σ-Δ mechanism

The density of "ones" at the modulator output is proportional to the input signal. For an increasing input the comparator generates a greater number of "ones," and vice versa for a decreasing input.
The average DAC output voltage is controlled by the *ones-density in the 1-bit data stream from the comparator output. As the input signal increases towards +VREF, the number of "ones" in the serial bit stream increases, and the number of "zeros" decreases. Similarly, as the signal goes negative towards –VREF, the number of "ones" in the serial bit stream decreases, and the number of "zeros" increases.*

For any given input value in a single sampling interval, the data from the 1-bit ADC is virtually meaningless. Only when a large number of samples are averaged, will a meaningful value result. The Σ-Δ modulator is very difficult to analyze in the time domain because of this apparent randomness of the single-bit data output. If the input signal is near positive full-scale, it is clear that there will be more "1"s than "0"s in the bit stream. Likewise, for signals near negative full-scale, there will be more "0"s than "1"s in the bit stream. For signals near midscale, there will be approximately an equal number of "1"s and "0"s.

From a very simplistic standpoint, this analysis shows that the average value of the input voltage is contained in the serial bit stream out of the comparator. The digital filter and decimator process the serial bit stream and produce the final output data

Sigma-Delta A/DConverters



Sigma-Delta Modulator Waveforms

Figure above shows the output of the integrator for two input conditions. The first is for an input of zero (midscale). To decode the output, pass the output samples through a simple digital lowpass filter that averages every four samples. The output of the filter is 2/4. This value represents bipolar zero. If more samples are averaged, more dynamic range is achieved.

The Converter waveforms for Vin=Vref/2 are also shown

Sigma-Delta Conversion Example

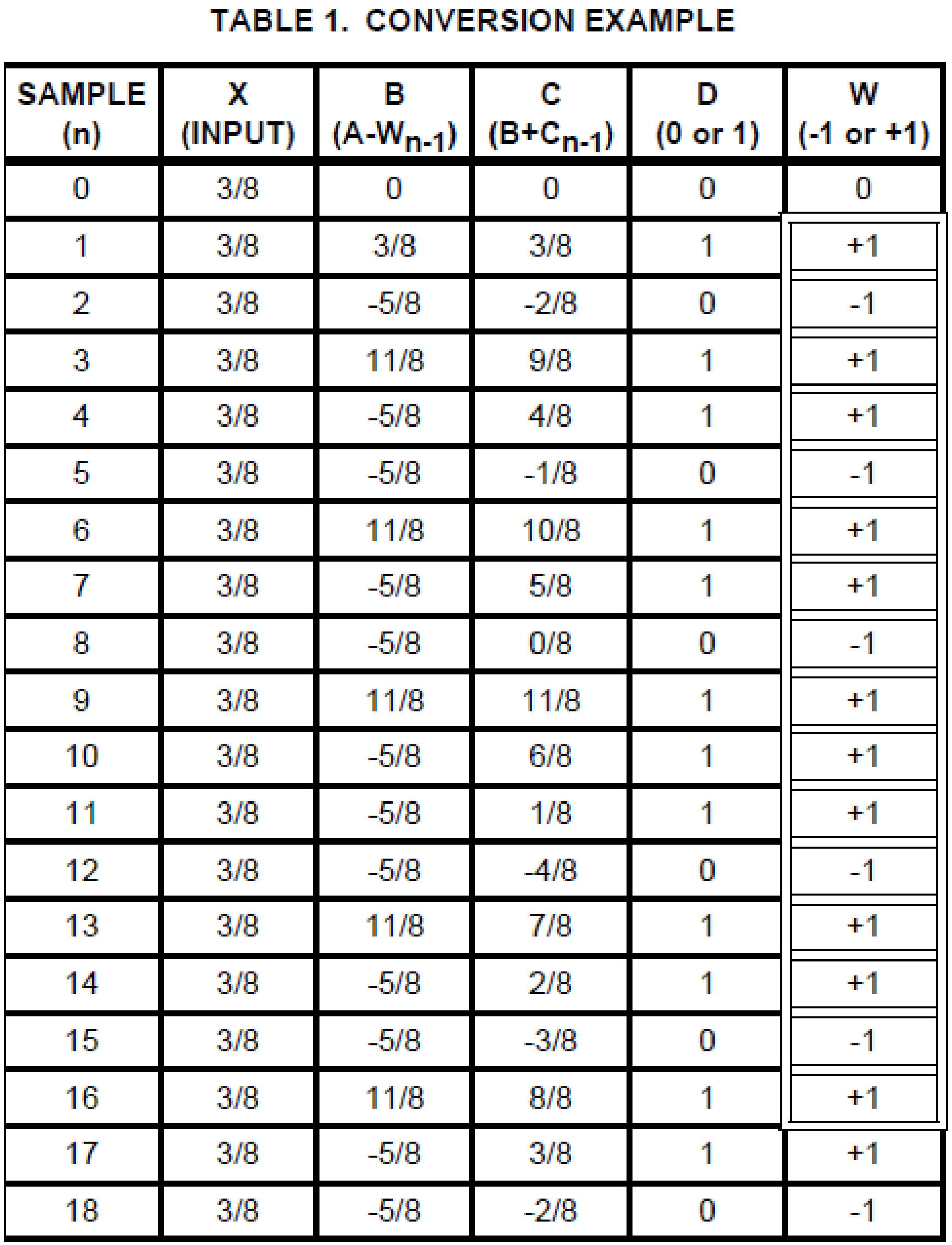
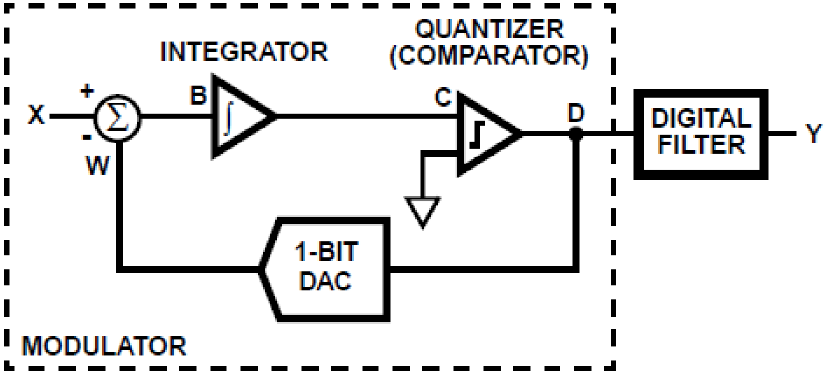


TABLE 1. CONVERSION EXAMPLE

SAMPLE (n)	X (INPUT)	B (A-W_{n-1})	C (B+C_{n-1})	D (0 or 1)	W (-1 or +1)
0	3/8	0	0	0	0
1	3/8	3/8	3/8	1	+1
2	3/8	-5/8	-2/8	0	-1
3	3/8	11/8	9/8	1	+1
4	3/8	-5/8	4/8	1	+1
5	3/8	-5/8	-1/8	0	-1
6	3/8	11/8	10/8	1	+1
7	3/8	-5/8	5/8	1	+1
8	3/8	-5/8	0/8	0	-1
9	3/8	11/8	11/8	1	+1
10	3/8	-5/8	6/8	1	+1
11	3/8	-5/8	1/8	1	+1
12	3/8	-5/8	-4/8	0	-1
13	3/8	11/8	7/8	1	+1
14	3/8	-5/8	2/8	1	+1
15	3/8	-5/8	-3/8	0	-1
16	3/8	11/8	8/8	1	+1
17	3/8	-5/8	3/8	1	+1
18	3/8	-5/8	-2/8	0	-1

It would be useful to show a quick conversion example. Referring to Table 1 the table headings X, B, C, D, and W correspond to points in the signal path of the block diagram of Figure 1 below.

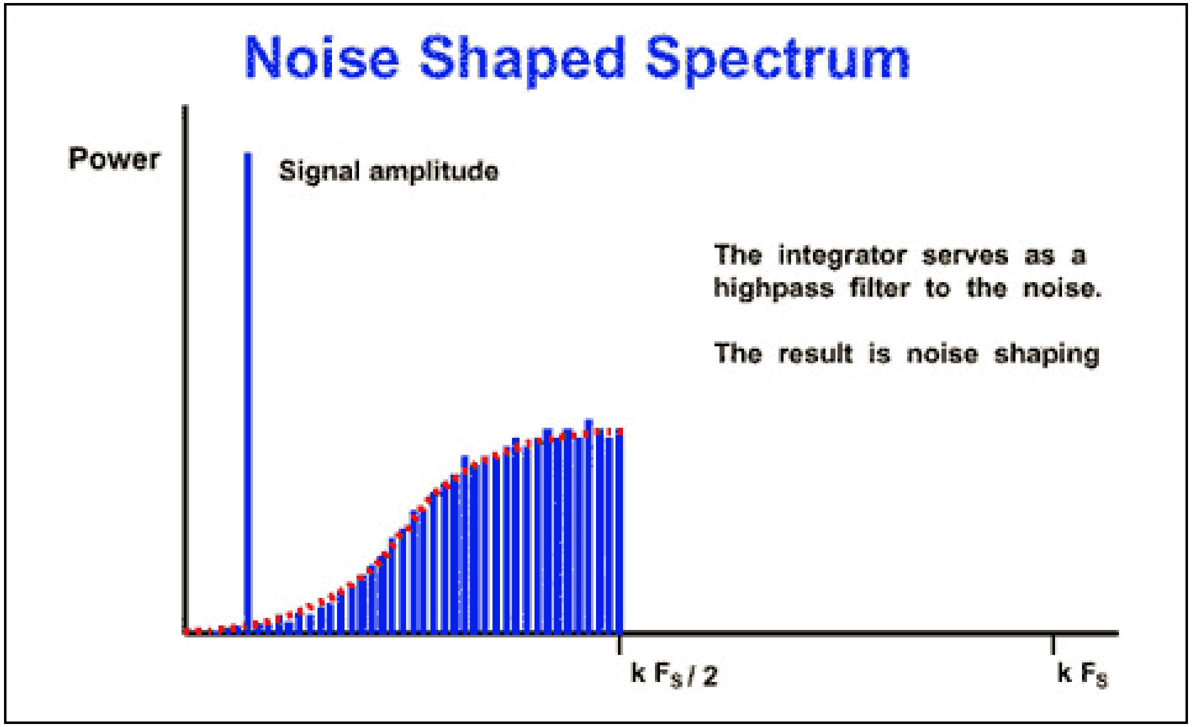
For this example the input X is a DC input of 3/8. The resultant signal at each point in the signal path for each signal sample is shown in Table 1. Note that a repetitive pattern develops every sixteen samples and that the average of the signal W over samples 1 to 16 is 3/8 thus showing that the feedback loop forces the average of the feedback signal W to be equal to the input X.



Sigma-Delta A/DConverters

Noise Shaping

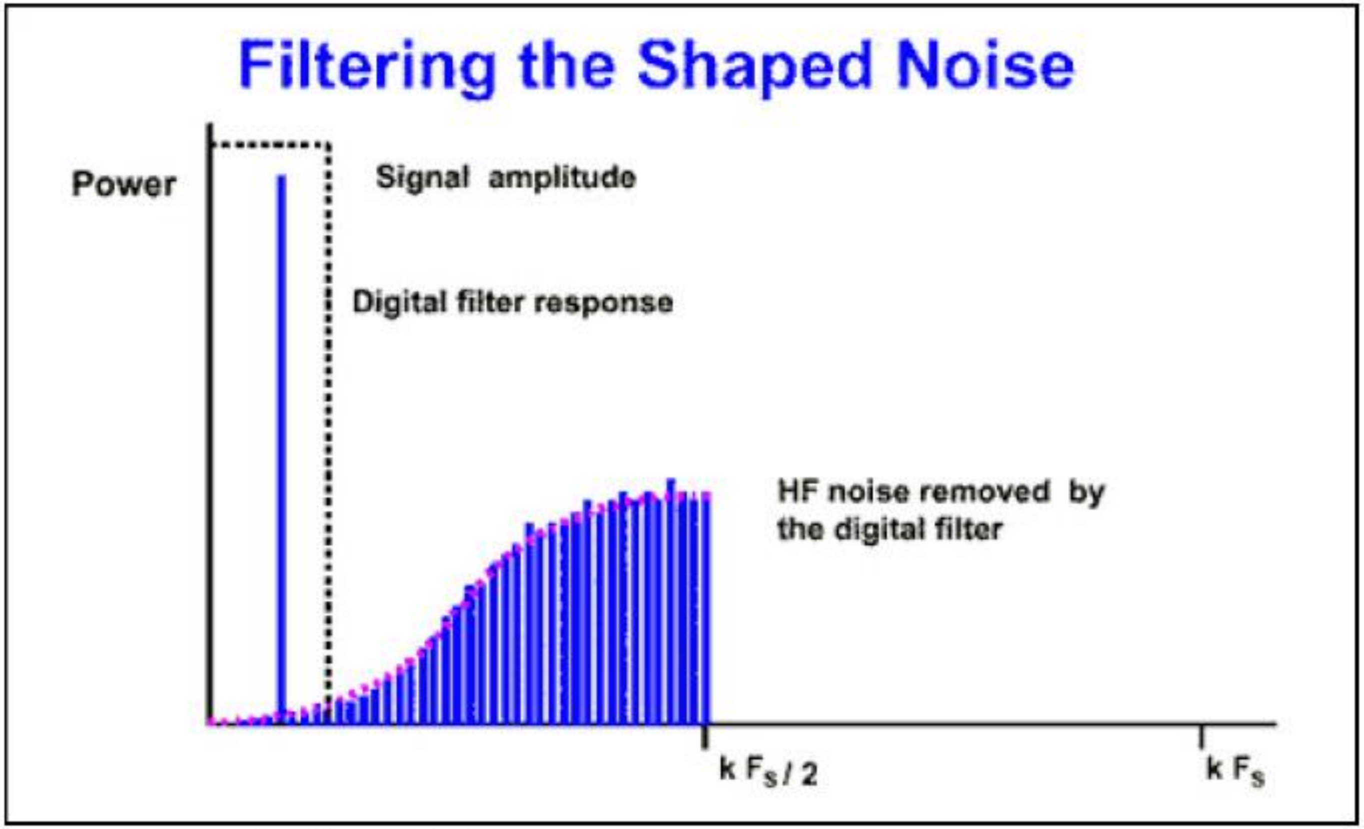
By summing the error voltage in the ΣΔ loop, the integrator acts as a lowpass filter to the input signal and a highpass filter to the quantization noise. Thus, most of the quantization noise is pushed into higher frequencies (**see Figure below)**



Effect of the integrator in the sigma-delta modulator.

Sigma-Delta A/DConverters

Noise Shaping : *Effect of the digital filter on the shaped noise*

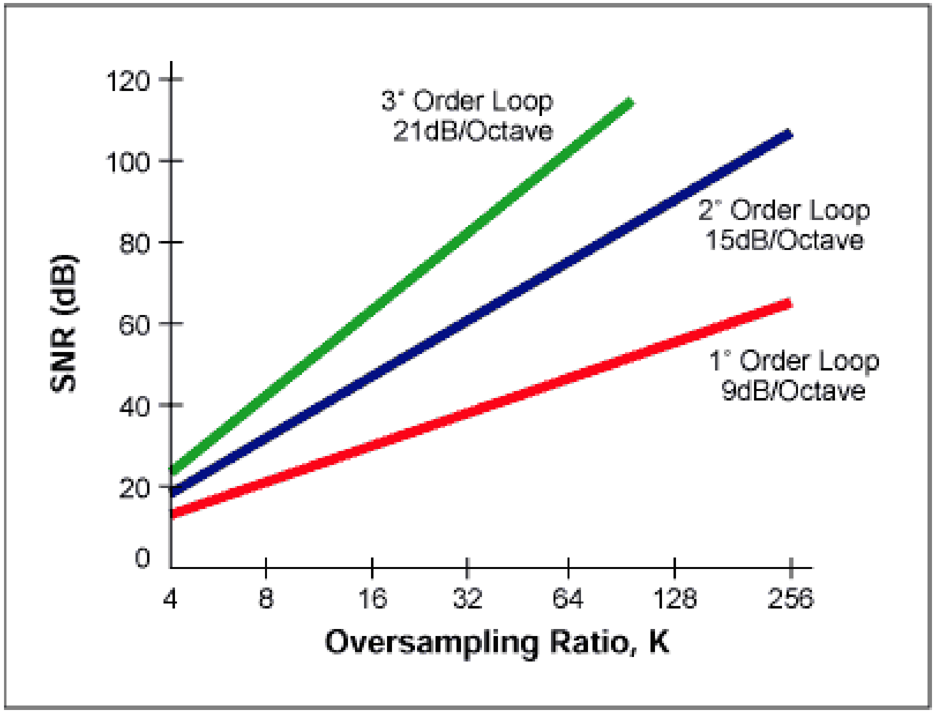


If we apply a digital filter to the noise-shaped delta-sigma modulator, it removes more noise than does simple oversampling .**This type of modulator (first-order) provides a 9dB improvement in SNR for every doubling of** the sampling rate.

For higher orders of quantization, we can achieve noise shaping by including more than one stage of integration and summing in the sigma-delta modulator.

For example, a second-order sigma-delta modulator **provides a 15dB improvement in SNR for every doubling of the sampling rate.**

Sigma-Delta A/DConverters



Relationship between order of sigma-delta modulator and the amount of over-sampling necessary to achieve a particular SNR.

The noise power in the bandwidth of interest for the 1st order modulator is

$$n_0^2 = e^2_{RMS}\frac{\pi^2}{3}\left(\frac{2f_0}{f_S}\right)^3$$

It can be shown that for the second order modulator the noise is

$$= e_{RMS}\left(\frac{\pi^2}{\sqrt{5}}\right)\left(\frac{2f_0}{f_S}\right)^{\frac{5}{2}}$$

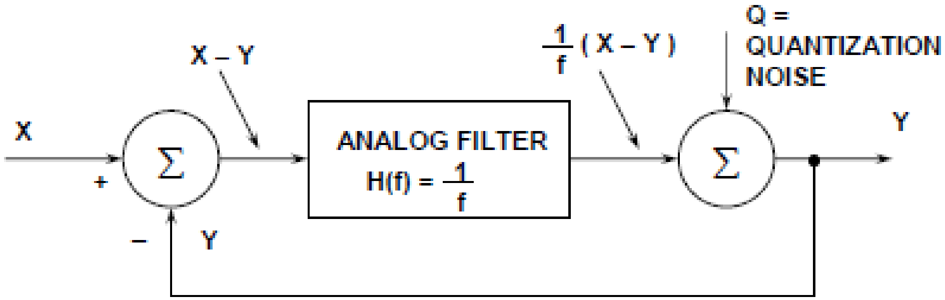
The generalized formula for the noise of an Mth order modulator is

$$n_0 = e_{RMS}\left(\frac{\pi^M}{\sqrt{2M+1}}\right)\left(\frac{2f_0}{f_S}\right)^{M+\frac{1}{2}}$$

and doubling the sampling frequency will decrease the inband quantization noise by 3(2M+1)dB.

Sigma-Delta A/DConverters

FREQUENCY DOMAIN ANALYSIS OF A SIGMA-DELTA ADC PROVING NOISE SHAPING



Simplified Frequency Domain Linearized Model of a Sigma-Delta Modulator

The integrator in the modulator is represented as an analog filter with a transfer function equal to H(f) = 1/f. The 1-bit quantizer generates quantization noise, Q, which is injected into the output summing block. If we let the input signal be X, and the output Y, the signal coming out of the input summer must be X – Y. This is multiplied by the filter transfer function, 1/f, and the result goes to one input of the output summer. By inspection, we can then write the expression for the output voltage Y as:

$$Y = \frac{1}{f}(X - Y) + Q.$$

This expression can easily be rearranged and solved for Y in terms of X, f, and Q

$$Y = \frac{X}{f+1} + \frac{Q \cdot f}{f+1}$$

Note that as the frequency f approaches zero, the output voltage Y approaches X with no noise component. At higher frequencies, the amplitude of the signal component approaches zero, and the noise component approaches Q. At high frequency, the output consists primarily of quantization noise. In essence, the analog filter has a lowpass effect on the signal, and a highpass effect on the quantization noise. Thus the analog filter performs the noise shaping function in the Σ-Δ modulator model offers more attenuation.

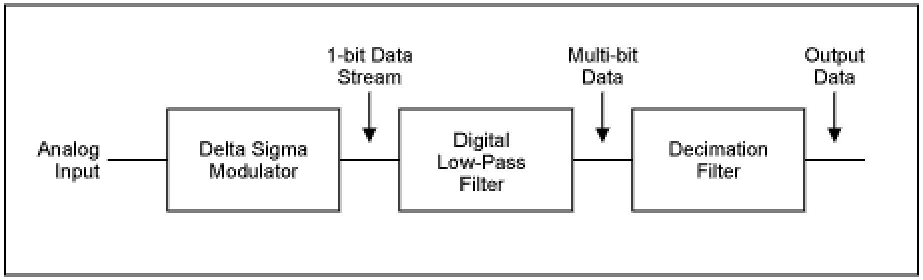
Sigma-Delta A/DConverters

Digital and Decimation Filter

The output of the sigma-delta modulator is a 1-bit data stream at the sampling rate, which can be in the megahertz range.

The purpose of the digital-and-decimation filter **is to extract information from this data stream and** reduce the data rate to a more useful value.

In a sigma-delta ADC, the digital filter averages the 1-bit data stream, improves the ADC resolution, and removes quantization noise that is outside the band of interest. It determines the signal bandwidth, settling time, and stopband rejection.



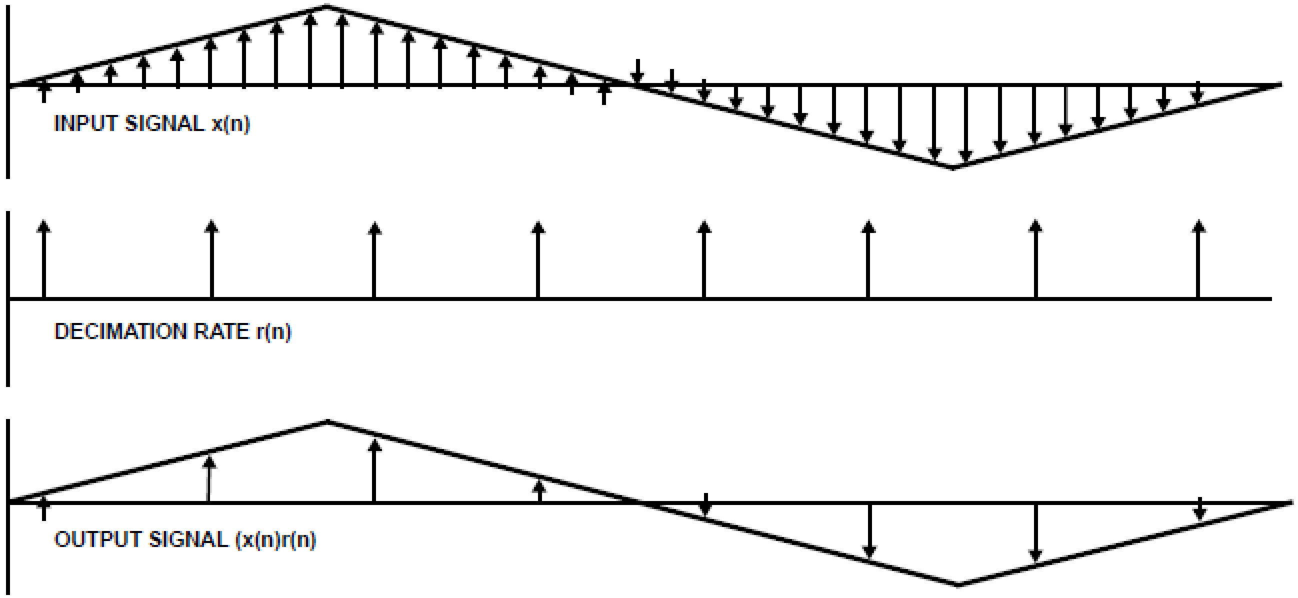
However, the digital filter does introduce inherent pipeline delay, which definitely must be considered in multiplexed and servo applications. If signals are multiplexed into a ΣΔ ADC, the digital filter must be allowed to settle to the new value before the output data is valid. Several output clock cycles are generally required for this settling. Because of the pipeline delay of the digital filter, the ΣΔ converter cannot be operated in a “single-shot” or “burst” mode.

For example, the group delay through a digital filter is 910 μs (sampling at 48 kSPS) and 460 μs (sampling at 96 kSPS)—this represents the time it takes for a step function input to propagate through one-half the number of taps in the digital filter. The total settling time is therefore approximately twice the group delay time. The input oversampling frequency is 6.144 MSPS for both conditions.

Sigma-Delta A/DConverters

Decimation

DECIMATION IN THE TIME DOMAIN



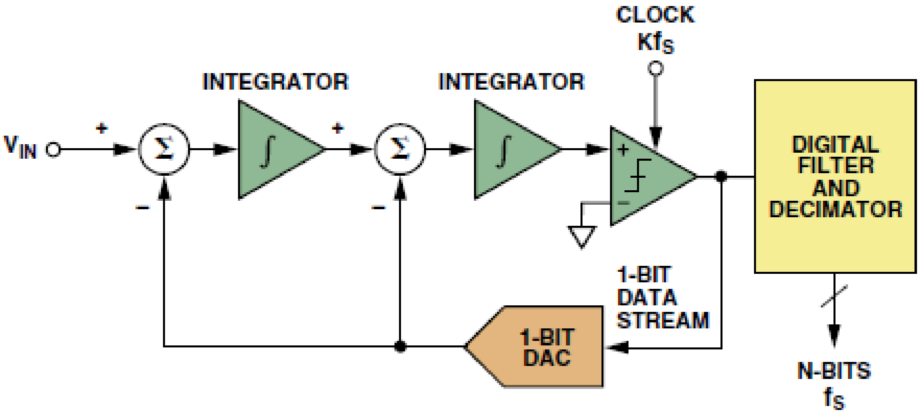
The process of decimation is used in a sigma delta converter to eliminate redundant data at the output. The sampling theorem tells us that the sample rate only needs to be 2 times the input signal bandwidth in order to reliably reconstruct the input signal without distortion.

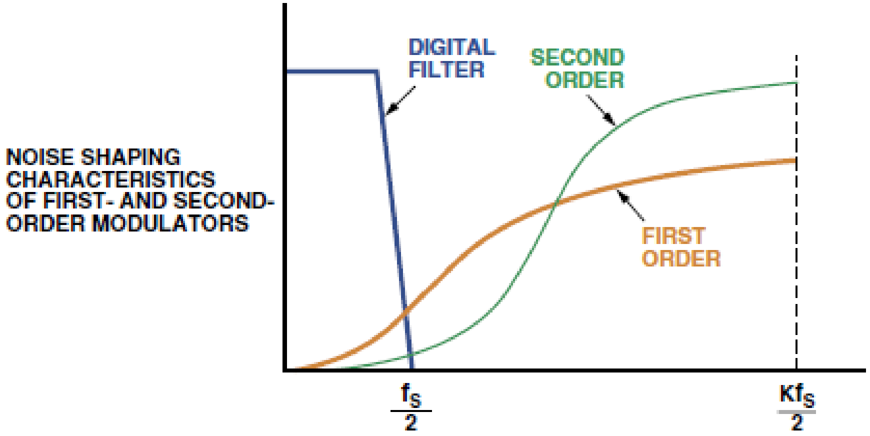
However, the input signal was grossly oversampled by the sigma delta modulator in order to reduce the quantization noise. Therefore, there is redundant data that can be eliminated without introducing distortion to the conversion result

Sigma-Delta A/DConverters

Although the simple first-order single-bit ΣΔ ADC is inherently linear and monotonic because of the 1-bit ADC and 1-bit DAC, it does not provide sufficient noise shaping for high-resolution applications.

Increasing the number of integrators in the modulator provides more noise shaping at the expense of a more complex design, as shown in Figure below for a second-order 1-bit modulator . Higher-order modulators (greater than third order) are difficult to stabilize and present significant design challenges.

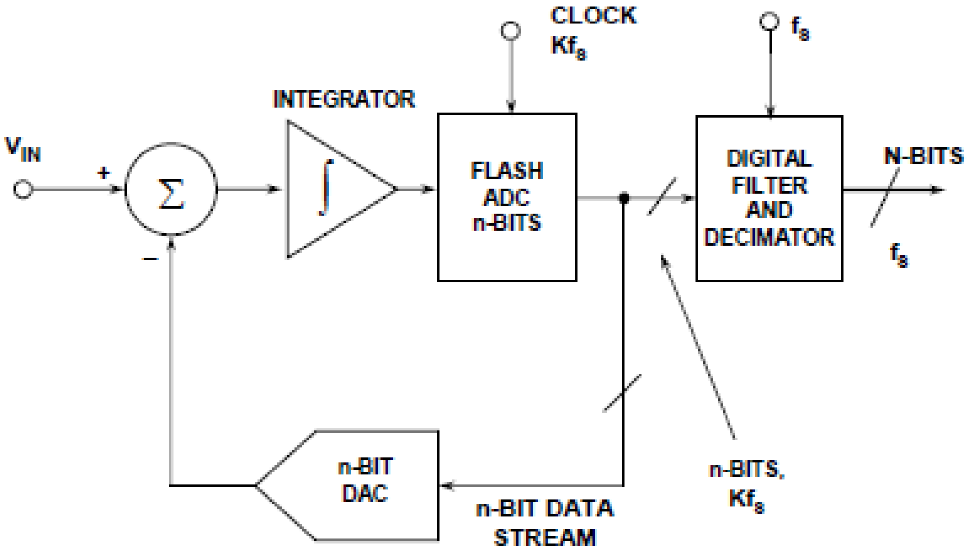




Sigma-Delta A/DConverters

MULTI-BIT SIGMA-DELTA CONVERTERS

The block diagram of Figure 5 shows a multi-bit Σ-Δ ADC which uses an n-bit flash ADC and an n-bit DAC. Obviously, this architecture will give a higher dynamic range for a given oversampling ratio and order of loop filter. Stabilization is easier, since second-order loops can generally be used.

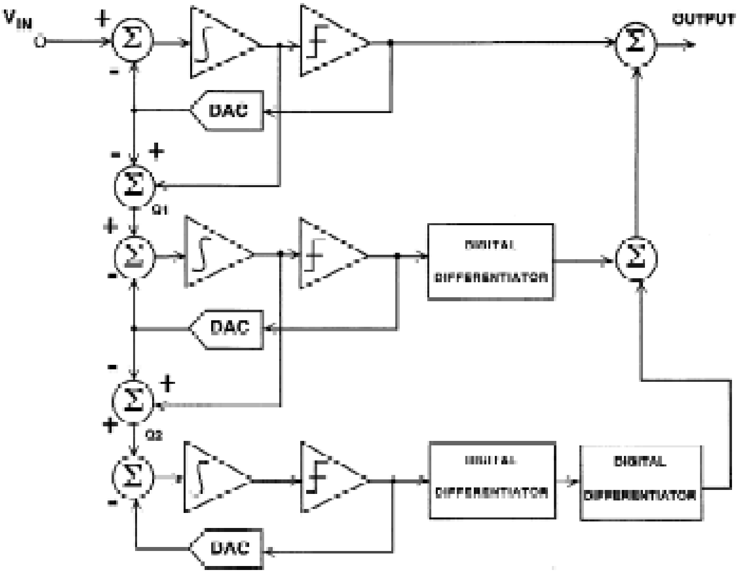


The real disadvantage of this technique is that the linearity depends on the DAC linearity, and special techniques, like thin film laser trimming is required to approach 16-bit performance levels. This makes the multi-bit architecture extremely impractical to implement on mixed-signal ICs using traditional binary DAC techniques.

Sigma-Delta A/DConverters

MULTISTAGE NOISE SHAPING (MASH) SIGMA-DELTA CONVERTERS

Nonlinear stabilization techniques can be difficult for 3rd order loops or higher. In many cases, the multi-bit architecture is preferable. An alternative approach to either of these, called multistage noise shaping (MASH), utilizes cascaded stable first-order loops Figure shows a block diagram of a three-stage MASH ADC. The output of the first integrator is subtracted from the first DAC output to yield the first stage quantization noise, Q1. Q1 is then quantized by the second stage. The output of the second integrator is subtracted from the second DAC output to yield the second stage quantization noise which is in turn quantized by the third stage



The output of the first stage is summed with a single digital differentiation of the second stage output and a double differentiation of the third stage output to yield the final output. The result is that the quantization noise Q1 is suppressed by the second stage, and the quantization noise Q2 is suppressed by the third stage yielding the same suppression as a third-order loop. Since this result is obtained using three first-order loops, stable operation is assured

Sigma-Delta A/DConverters

SUMMARY

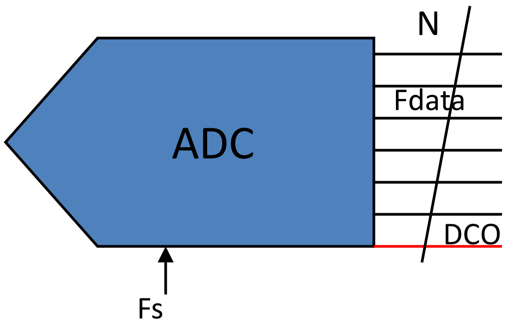
Sigma-delta ADCs and DACs have proliferated into many modern applications including measurement, voiceband, audio, etc. The technique takes full advantage of low cost CMOS processes and therefore makes integration with highly digital functions such as DSPs practical.

Resolutions up to 24-bits are currently available, and the requirements on analog antialiasing/anti-imaging filters are greatly relaxed due to oversampling. The internal digital filter in audio Σ-Δ ADCs can be designed for linear phase, which is a major requirement in those applications.

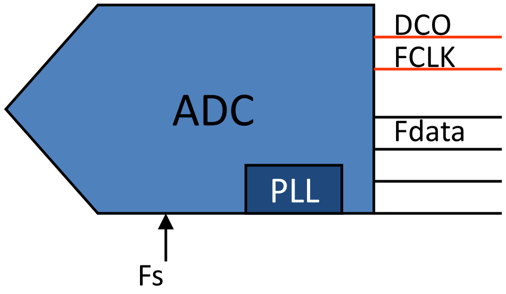
Many Σ-Δ converters offer a high level of user programmability with respect to output data rate, digital filter characteristics, and self-calibration modes. Multi-channel Σ-Δ ADCs are now available for data acquisition systems, and most users are well-educated with respect to the settling time requirements of the internal digital filter in these applications.

ADC Output Configurations

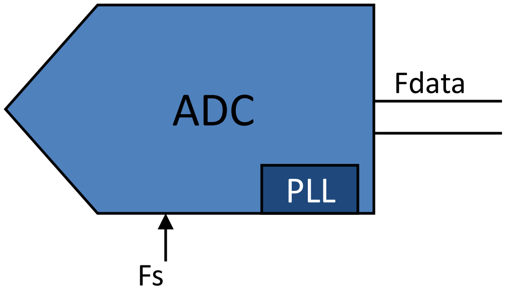
PARALLEL



SERIAL LVDS



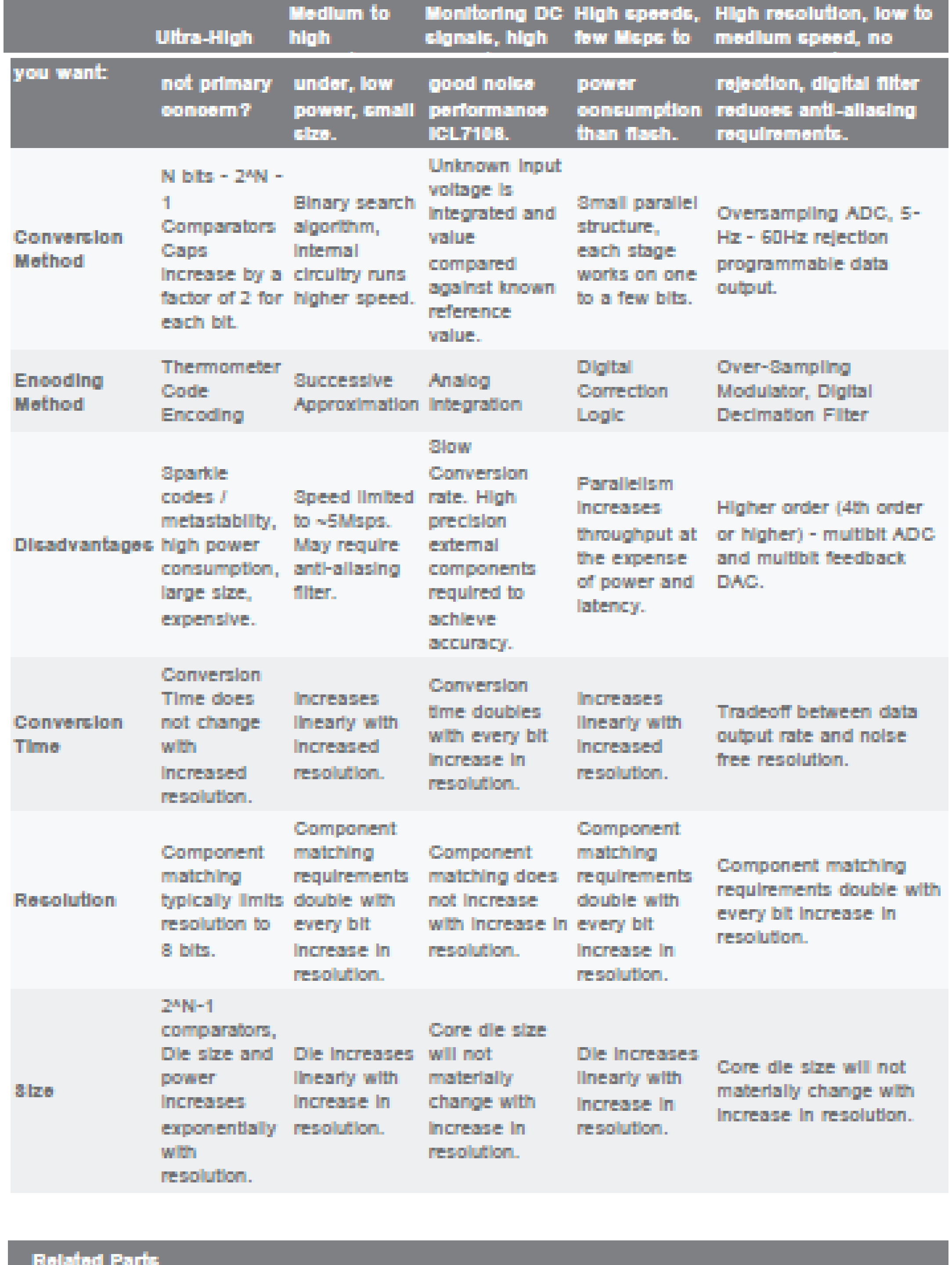
SerDes



- Parallel CMOS
  - F_{data} max = 150 MSPS
- DDR LVDS
  - F_{data} max = 420 MSPS
- Interface available in lower cost FPGAs
- Pins = ADC resolution plus DCO
- High pin count

- F_{data} max = 840Mbps?
- Serial LVDS
  - F_s max = F_{data} * # of data lanes / ADC resolution
- On chip PLL required
- Higher-end FPGA typically required
- Pins = # of data lanes plus Frame CLK and Data CLK

- F_{data} = 3.125Gbps+
  - Encoded serial CML
  - F_s max = data packet length + overhead
- On chip PLL required
- High-end FPGA required
  - Clock recovery
- Slower customer adoption rate
- 2 pins



	Ultra-High	Medium to high	Monitoring DC signals, high	High speeds, few Msps to	High resolution, low to medium speed, no
you want:	not primary concern?	under, low power, small size.	good noise performance ICL7106.	power consumption than flash.	rejection, digital filter reduces anti-aliasing requirements.
Conversion Method	N bits - 2^N - 1 Comparators Caps increase by a factor of 2 for each bit.	Binary search algorithm, internal circuitry runs higher speed.	Unknown input voltage is integrated and value compared against known reference value.	Small parallel structure, each stage works on one to a few bits.	Oversampling ADC, 5-Hz - 60Hz rejection programmable data output.
Encoding Method	Thermometer Code Encoding	Successive Approximation	Analog Integration	Digital Correction Logic	Over-Sampling Modulator, Digital Decimation Filter
Disadvantages	Sparkle codes / metastability, high power consumption, large size, expensive.	Speed limited to ~5Msps. May require anti-aliasing filter.	Slow Conversion rate. High precision external components required to achieve accuracy.	Parallelism increases throughput at the expense of power and latency.	Higher order (4th order or higher) - multibit ADC and multibit feedback DAC.
Conversion Time	Conversion Time does not change with increased resolution.	Increases linearly with increased resolution.	Conversion time doubles with every bit increase in resolution.	Increases linearly with increased resolution.	Tradeoff between data output rate and noise free resolution.
Resolution	Component matching typically limits resolution to 8 bits.	Component matching requirements double with every bit increase in resolution.	Component matching does not increase with increase in resolution.	Component matching requirements double with every bit increase in resolution.	Component matching requirements double with every bit increase in resolution.
Size	2^N-1 comparators, Die size and power increases exponentially with resolution.	Die increases linearly with increase in resolution.	Core die size will not materially change with increase in resolution.	Die increases linearly with increase in resolution.	Core die size will not materially change with increase in resolution.

Related Parts