Digital-to-Analog and Analog-to-Digital Converters

M. B. Patil
mbpatil@ee.iitb.ac.in
www.ee.iitb.ac.in/~sequel

Department of Electrical Engineering
Indian Institute of Technology Bombay
Introduction

Real signals (e.g., a voltage measured with a thermocouple or a speech signal recorded with a microphone) are analog quantities, varying continuously with time. Digital format offers several advantages: digital signal processing, storage, use of computers, robust transmission, etc.

An ADC (Analog-to-Digital Converter) is used to convert an analog signal to the digital format. The reverse conversion (from digital to analog) is also required. For example, music stored in a DVD in digital format must be converted to an analog voltage for playing out on a speaker.

A DAC (Digital-to-Analog Converter) is used to convert a digital signal to the analog format.
Real signals (e.g., a voltage measured with a thermocouple or a speech signal recorded with a microphone) are analog quantities, varying continuously with time.
* Real signals (e.g., a voltage measured with a thermocouple or a speech signal recorded with a microphone) are analog quantities, varying continuously with time.

* Digital format offers several advantages: digital signal processing, storage, use of computers, robust transmission, etc.
* Real signals (e.g., a voltage measured with a thermocouple or a speech signal recorded with a microphone) are analog quantities, varying continuously with time.

* Digital format offers several advantages: digital signal processing, storage, use of computers, robust transmission, etc.

* An ADC (Analog-to-Digital Converter) is used to convert an analog signal to the digital format.
* Real signals (e.g., a voltage measured with a thermocouple or a speech signal recorded with a microphone) are analog quantities, varying continuously with time.

* Digital format offers several advantages: digital signal processing, storage, use of computers, robust transmission, etc.

* An ADC (Analog-to-Digital Converter) is used to convert an analog signal to the digital format.

* The reverse conversion (from digital to analog) is also required. For example, music stored in a DVD in digital format must be converted to an analog voltage for playing out on a speaker.
* Real signals (e.g., a voltage measured with a thermocouple or a speech signal recorded with a microphone) are analog quantities, varying continuously with time.
* Digital format offers several advantages: digital signal processing, storage, use of computers, robust transmission, etc.
* An ADC (Analog-to-Digital Converter) is used to convert an analog signal to the digital format.
* The reverse conversion (from digital to analog) is also required. For example, music stored in a DVD in digital format must be converted to an analog voltage for playing out on a speaker.
* A DAC (Digital-to-Analog Converter) is used to convert a digital signal to the analog format.
For a 4-bit DAC, with input $S_3 S_2 S_1 S_0$, the output voltage is $V_A = K \left[ (S_3 \times 2^3) + (S_2 \times 2^2) + (S_1 \times 2^1) + (S_0 \times 2^0) \right]$. 

In general, $V_A = K \sum_{N-1}^{0} S_k 2^k$.

*K is proportional to the reference voltage $V_R$. Its value depends on how the DAC is implemented.*

M. B. Patil, IIT Bombay
For a 4-bit DAC, with input $S_3S_2S_1S_0$, the output voltage is

$$V_A = K \left[(S_3 \times 2^3) + (S_2 \times 2^2) + (S_1 \times 2^1) + (S_0 \times 2^0)\right].$$

In general, $V_A = K \sum_{0}^{N-1} S_k 2^k$. 
For a 4-bit DAC, with input $S_3S_2S_1S_0$, the output voltage is

$$V_A = K \left[ (S_3 \times 2^3) + (S_2 \times 2^2) + (S_1 \times 2^1) + (S_0 \times 2^0) \right].$$

In general, $V_A = K \sum_{0}^{N-1} S_k 2^k$. 
For a 4-bit DAC, with input $S_3S_2S_1S_0$, the output voltage is

$$V_A = K \left[ (S_3 \times 2^3) + (S_2 \times 2^2) + (S_1 \times 2^1) + (S_0 \times 2^0) \right].$$

In general, $V_A = K \sum_{0}^{N-1} S_k 2^k$.

* $K$ is proportional to the reference voltage $V_R$. Its value depends on how the DAC is implemented.
DAC using binary-weighted resistors

Inputs: \( S_3, S_2, S_1, S_0 \)
Output: \( V_A \)

\[ V_R \]

\[ \begin{align*}
R_3 &= R \\
R_2 &= 2R \\
R_1 &= 4R \\
R_0 &= 8R
\end{align*} \]

\[ R_f \]

\[ \begin{align*}
A_3 &\quad I_3 \\
A_2 &\quad I_2 \\
A_1 &\quad I_1 \\
A_0 &\quad I_0
\end{align*} \]

\[ V_A \]

\[ \begin{align*}
S_3 V_R &\quad A_3 \\
S_2 V_R &\quad A_2 \\
S_1 V_R &\quad A_1 \\
S_0 V_R &\quad A_0
\end{align*} \]

\[ R_f \]

\[ \begin{align*}
I &= S_0 V_R + S_1 V_R + S_2 V_R + S_3 V_R \\
&= \sum_{k=0}^{N-1} S_k \times 2^k (N=4)
\end{align*} \]

\[ V_o = -R_f I = -V_R R_f 2^{N-1} \sum_{k=0}^{N-1} S_k \times 2^k \]
DAC using binary-weighted resistors

**Inputs:** $S_3, S_2, S_1, S_0$

**Output:** $V_A$

* If the input bit $S_k$ is 1, $A_k$ gets connected to $V_R$; else, it gets connected to ground.

M. B. Patil, IIT Bombay
DAC using binary-weighted resistors

Inputs: $S_3, S_2, S_1, S_0$
Output: $V_A$

* If the input bit $S_k$ is 1, $A_k$ gets connected to $V_R$; else, it gets connected to ground. → $V(A_k) = S_k \times V_R$.
DAC using binary-weighted resistors

Inputs: \( S_3, S_2, S_1, S_0 \)
Output: \( V_A \)

\[ R_3 = R \]
\[ R_2 = 2R \]
\[ R_1 = 4R \]
\[ R_0 = 8R \]

\[ I_0 = \frac{V(A_0) - 0}{R_0} = \frac{S_0 \cdot V_R}{R_0} \]

\[ I_1 = \frac{V(A_1) - 0}{R_1} = \frac{S_1 \cdot V_R}{R_1} \]

\[ I_2 = \frac{V(A_2) - 0}{R_2} = \frac{S_2 \cdot V_R}{R_2} \]

\[ I_3 = \frac{V(A_3) - 0}{R_3} = \frac{S_3 \cdot V_R}{R_3} \]

* If the input bit \( S_k \) is 1, \( A_k \) gets connected to \( V_R \); else, it gets connected to ground. \( \rightarrow V(A_k) = S_k \times V_R \).

* Since the inverting terminal of the op-amp is at virtual ground, \( I_k = \frac{V(A_k) - 0}{R_k} = \frac{S_k \cdot V_R}{R_k} \).
DAC using binary-weighted resistors

Inputs: \( S_3, S_2, S_1, S_0 \)
Output: \( V_A \)

\[
\begin{align*}
R_0 &= 8R \\
R_1 &= 4R \\
R_2 &= 2R \\
R_3 &= R \\
R_f &= \frac{V_R}{I_0} \\
V_A &= -R_f I
\end{align*}
\]

* If the input bit \( S_k \) is 1, \( A_k \) gets connected to \( V_R \); else, it gets connected to ground. \( \rightarrow V(A_k) = S_k \times V_R \).

* Since the inverting terminal of the op-amp is at virtual ground, \( I_k = \frac{V(A_k) - 0}{R_k} = \frac{S_k \times V_R}{R_k} \).

\[
I = \frac{S_0 \times V_R}{8R} + \frac{S_1 \times V_R}{4R} + \frac{S_2 \times V_R}{2R} + \frac{S_3 \times V_R}{R} = \frac{V_R}{2^{N-1}R} \sum_{0}^{N-1} S_k \times 2^k \quad (N = 4).
\]
DAC using binary-weighted resistors

* If the input bit $S_k$ is 1, $A_k$ gets connected to $V_R$; else, it gets connected to ground. \( \rightarrow V(A_k) = S_k \times V_R. \)

* Since the inverting terminal of the op-amp is at virtual ground, $I_k = \frac{V(A_k) - 0}{R_k} = \frac{S_k \times V_R}{R_k}$.

* $I = \frac{S_0 V_R}{8R} + \frac{S_1 V_R}{4R} + \frac{S_2 V_R}{2R} + \frac{S_3 V_R}{R} = \frac{V_R}{2^{N-1}R} \sum_{0}^{N-1} S_k \times 2^k \ (N = 4)$.

* The output voltage is $V_o = -R_f I = -V_R \frac{R_f}{2^{N-1}R} \sum_{0}^{N-1} S_k \times 2^k$. 

M. B. Patil, IIT Bombay
Consider an 8-bit DAC with $V_R = 5\, \text{V}$. What is the smallest value of $R$ which will limit the current drawn from the supply ($V_R$) to 10 mA?

Maximum current is drawn from $V_R$ when the input is $1111\,1111$. → All nodes $A_0$ to $A_7$ get connected to $V_R$.

$10\, \text{mA} = V_R \frac{R}{2^0} + V_R \frac{R}{2^1} + \cdots + V_R \frac{R}{2^7} = \frac{1}{2^7} V_R R (2^8 - 1) = 255 R$.

$\Rightarrow R_{\text{min}} = \frac{5\, \text{V}}{10\, \text{mA} \times 255} = 996\, \Omega$.

Consider an 8-bit DAC with $V_R = 5$ V. What is the smallest value of $R$ which will limit the current drawn from the supply ($V_R$) to 10 mA?
Consider an 8-bit DAC with $V_R = 5\,\text{V}$. What is the smallest value of $R$ which will limit the current drawn from the supply ($V_R$) to 10 mA?

Maximum current is drawn from $V_R$ when the input is 1111 1111.
* Consider an 8-bit DAC with \( V_R = 5 \text{ V} \). What is the smallest value of \( R \) which will limit the current drawn from the supply \((V_R)\) to 10 mA?

Maximum current is drawn from \( V_R \) when the input is 1111 1111.
→ All nodes \( A_0 \) to \( A_7 \) get connected to \( V_R \).
Consider an 8-bit DAC with $V_R = 5 \text{ V}$. What is the smallest value of $R$ which will limit the current drawn from the supply ($V_R$) to 10 mA?

Maximum current is drawn from $V_R$ when the input is 1111 1111.

$\rightarrow$ All nodes $A_0$ to $A_7$ get connected to $V_R$.

$\rightarrow$ \[ 10 \text{ mA} = \frac{V_R}{R} + \frac{V_R}{2R} + \cdots + \frac{V_R}{2^7R} = \frac{1}{2^7} \frac{V_R}{R} \left( 2^0 + 2^1 + \cdots + 2^7 \right) \]

\[ = \frac{1}{2^7} \frac{V_R}{R} \left( 2^8 - 1 \right) = \frac{255}{128} \frac{V_R}{R} \]
Consider an 8-bit DAC with $V_R = 5\ \text{V}$. What is the smallest value of $R$ which will limit the current drawn from the supply ($V_R$) to $10\ \text{mA}$?

Maximum current is drawn from $V_R$ when the input is 1111 1111.

→ All nodes $A_0$ to $A_7$ get connected to $V_R$.

→ $10\ \text{mA} = \frac{V_R}{R} + \frac{V_R}{2R} + \cdots + \frac{V_R}{2^7R} = \frac{1}{2^7} \frac{V_R}{R} \left(2^0 + 2^1 + \cdots + 2^7\right)$

\[
= \frac{1}{2^7} \frac{V_R}{R} \left(2^8 - 1\right) = \frac{255}{128} \frac{V_R}{R} \rightarrow R_{\text{min}} = \frac{5\ \text{V}}{10\ \text{mA}} \times 255 \frac{128}{2^7} = 996\ \Omega .
\]

DAC using binary-weighted resistors: Example (from Gopalan)

\[ V_A \]

\[ R_f \]

\[ I_7 \]

\[ I_1 \]

\[ I_0 \]

\[ A_7 \]

\[ A_1 \]

\[ A_0 \]

\[ R_7 = R \]

\[ R_1 = 2^6 R \]

\[ R_0 = 2^7 R \]

\[ V_R \]

\[ M. B. Patil, IIT Bombay \]

If \( R_f = R \), what is the resolution (i.e., \( \Delta V_A \) corresponding to the input LSB changing from 0 to 1 with other input bits constant)?

\[ V_A = \frac{-VR}{R_f} \]

\[ \Delta V_A = V_A \left( \frac{R_f}{2^{N-1}} \right) = \frac{5V}{2^8} - 1 \times \frac{1}{2} = \frac{5}{128} \approx 0.0391 \, V \]
DAC using binary-weighted resistors: Example (from Gopalan)

If $R_f = R$, what is the resolution (i.e., $\Delta V_A$ corresponding to the input LSB changing from 0 to 1 with other input bits constant)?

* $\Delta V_A = \frac{1}{2^{N-1}} \times \frac{1}{R_f} = \frac{5}{2^8 - 1} \times 1 = 0.0391$ V.

M. B. Patil, IIT Bombay
* If $R_f = R$, what is the resolution (i.e., $\Delta V_A$ corresponding to the input LSB changing from 0 to 1 with other input bits constant)?

$$V_A = -V_R \frac{R_f}{2^{N-1}R} \left[ S_7 2^7 + \cdots + S_1 2^1 + S_0 2^0 \right]$$
DAC using binary-weighted resistors: Example (from Gopalan)

* If $R_f = R$, what is the resolution (i.e., $\Delta V_A$ corresponding to the input LSB changing from 0 to 1 with other input bits constant)?

$$V_A = -V_R \frac{R_f}{2^{N-1}R} \left[ S_7 2^7 + \cdots + S_1 2^1 + S_0 2^0 \right]$$

$$\rightarrow \Delta V_A = \frac{V_R}{2^{N-1}} \frac{R_f}{R} = \frac{5V}{2^{8-1}} \times 1 = \frac{5}{128} = 0.0391 \text{ V.}$$
What is the maximum output voltage (in magnitude)?

\[
V_A = -V_R \left( \frac{2^N - 1}{R_f R} \right) \left( 2^{A_1} + 2^{A_0} + \cdots + 2^0 \right).
\]

Maximum \( |V_A| \) is obtained when the input is \( 1111\ 1111 \).

\[
|V_A|_{\text{max}} = 5 \times 255_{128} = 9.961 \text{ V}.
\]
* What is the maximum output voltage (in magnitude)?
What is the maximum output voltage (in magnitude)?

\[ V_A = -\frac{V_R}{2^{N-1}} \frac{R_f}{R} \left[ S_72^7 + \cdots + S_12^1 + S_02^0 \right]. \]
What is the maximum output voltage (in magnitude)?

\[ V_A = -\frac{V_R}{2^{N-1}} \frac{R_f}{R} \left[ S_7 2^7 + \cdots + S_1 2^1 + S_0 2^0 \right]. \]

Maximum \( V_A \) (in magnitude) is obtained when the input is 1111 1111.

M. B. Patil, IIT Bombay
DAC using binary-weighted resistors: Example (from Gopalan)

\[ V_A = -\frac{V_R}{2^{N-1}} \frac{R_f}{R} \left[ S_7 2^7 + \cdots + S_1 2^1 + S_0 2^0 \right]. \]

Maximum \( V_A \) (in magnitude) is obtained when the input is 1111 1111.

\[ |V_A|_{\text{max}} = \frac{5}{128} \times 1 \times \left[ 2^0 + 2^1 + \cdots + 2^7 \right] = \frac{5}{128} \times \left( 2^8 - 1 \right) = 5 \times \frac{255}{128} = 9.961 \text{ V}. \]
DAC using binary-weighted resistors: Example (from Gopalan)

\[ V_A = -V_R \frac{R_f}{2^N - 1} \left[ S_7 \frac{2^7 R}{R_f} + \cdots + S_1 \frac{2^6 R}{R_f} + S_0 \frac{2^0 R}{R_f} \right]. \]

\[ = -\frac{5}{128} \times 1 \times \left[ \frac{2^7}{R_f} + \frac{2^6}{R_f} + \frac{2^5}{R_f} + \frac{2^4}{R_f} + \frac{2^0}{R_f} \right] = \frac{-5 \times 173}{128} = -6.758 V. \]
* Find the output voltage corresponding to the input 1010 1101.
* Find the output voltage corresponding to the input 1010 1101.

\[ V_A = - \frac{V_R}{2^{N-1}} \frac{R_f}{R} \left[ S_7 2^7 + \cdots + S_1 2^1 + S_0 2^0 \right]. \]
* Find the output voltage corresponding to the input 1010 1101.

\[
V_A = -\frac{V_R}{2^{N-1}} \frac{R_f}{R} \left[ S_7 2^7 + \cdots + S_1 2^1 + S_0 2^0 \right].
\]

\[
= -\frac{5}{128} \times 1 \times \left[ 2^7 + 2^5 + 2^3 + 2^2 + 2^0 \right] = -5 \times \frac{173}{128} = -6.758 \text{ V}.
\]
DAC using binary-weighted resistors: Example (from Gopalan)

If the resistors are specified to have a tolerance of 1%, what is the range of $|V_A|$ corresponding to input $1111\ 1111$?

$|V_A|$ is maximum when (a) currents $I_0$, $I_1$, etc. assume their maximum values, with $R_k = R_0 k (1 - 0.01)$ and (b) $R_f$ is maximum, $R_f = R_0 f (1 + 0.01)$.

(The superscript '0' denotes nominal value.)

$$|V_A|_{\text{max}}^{1111\ 1111} = V_R \times 2^{128} R_f |R|_{\text{max}} = 5 \times 2^{128} R_f \times 1.01 = 10.162 \text{ V}$$

Similarly, $|V_A|_{\text{min}}^{1111\ 1111} = 5 \times 2^{128} R_f \times 0.99 = 9.764 \text{ V}$.

M. B. Patil, IIT Bombay
* If the resistors are specified to have a tolerance of 1\% , what is the range of $|V_A|$ corresponding to input 1111 1111?
* If the resistors are specified to have a tolerance of 1%, what is the range of $|V_A|$ corresponding to input 1111 1111?

$|V_A|$ is maximum when (a) currents $I_0$, $I_1$, etc. assume their maximum values, with $R_k = R_k^0 \times (1 - 0.01)$ and (b) $R_f$ is maximum, $R_f = R_f^0 \times (1 + 0.01)$.

(The superscript ‘0’ denotes nominal value.)
DAC using binary-weighted resistors: Example (from Gopalan)

If the resistors are specified to have a tolerance of 1%, what is the range of $|V_A|$ corresponding to input 1111 1111?

$|V_A|$ is maximum when (a) currents $I_0$, $I_1$, etc. assume their maximum values, with $R_k = R_k^0 \times (1 - 0.01)$ and (b) $R_f$ is maximum, $R_f = R_f^0 \times (1 + 0.01)$.

(The superscript ‘0’ denotes nominal value.)

$$\rightarrow |V_A|_{11111111}^{\text{max}} = V_R \times \frac{255}{128} \times \frac{R_f}{R} \Bigg|^{\text{max}} = 5 \times \frac{255}{128} \times \frac{1.01}{0.99} = 10.162 \text{ V}.$$
DAC using binary-weighted resistors: Example (from Gopalan)

* If the resistors are specified to have a tolerance of 1%, what is the range of $|V_A|$ corresponding to input 1111 1111?

$|V_A|$ is maximum when (a) currents $I_0$, $I_1$, etc. assume their maximum values, with $R_k = R_k^0 \times (1 - 0.01)$ and (b) $R_f$ is maximum, $R_f = R_f^0 \times (1 + 0.01)$.

(The superscript ‘0’ denotes nominal value.)

$|V_A|_{11111111}^{\text{max}} = V_R \times \frac{255}{128} \times \frac{R_f}{R} \bigg|^{\text{max}} = 5 \times \frac{255}{128} \times \frac{1.01}{0.99} = 10.162 \text{ V.}$

Similarly, $|V_A|_{11111111}^{\text{min}} = 5 \times \frac{255}{128} \times \frac{0.99}{1.01} = 9.764 \text{ V.}$
DAC using binary-weighted resistors: Example (from Gopalan)

\[ V_A = \frac{R}{2^7 R} \]

\[ I_0 = \frac{2^6 R}{R_0} \]

\[ I_1 = \frac{2^5 R}{R_1} \]

\[ I_7 = \frac{R}{R_7} \]

\[ I = I_0 + I_1 + I_7 \]

\[ \Delta V_A \text{ for input } 1111 \ 1111 = 10.162 - 9.764 \approx 0.4 \text{ V} \]

This situation is not acceptable. The output voltage variation can be reduced by using resistors with a smaller tolerance. However, it is difficult to fabricate an IC with widely varying resistance values (from \( R \) to \( 2^{N-1} R \)) and each with a small enough tolerance. Use \( R - 2R \) ladder network instead.

M. B. Patil, IIT Bombay
DAC using binary-weighted resistors: Example (from Gopalan)

\[ R_7 = R \]

\[ R_1 = 2^6 R \]

\[ R_0 = 2^7 R \]

\[ \Delta V_A \text{ for input } 1111 
1111 = 10.162 - 9.764 \approx 0.4 \text{ V} \] which is larger than the resolution (0.039 V) of the DAC. This situation is not acceptable.
DAC using binary-weighted resistors: Example (from Gopalan)

\[
\begin{align*}
V_A & = R_{f} \\
I_7 & = R/V_R \\
I_1 & = 2^6 R/I_7 \\
I_0 & = 2^7 R/I_1
\end{align*}
\]

* $\Delta V_A$ for input 1111 1111 = 10.162 – 9.764 ≈ 0.4 V which is larger than the resolution (0.039 V) of the DAC. This situation is not acceptable.

* The output voltage variation can be reduced by using resistors with a smaller tolerance. However, it is difficult to fabricate an IC with widely varying resistance values (from $R$ to $2^{N-1} R$) and each with a small enough tolerance.
DAC using binary-weighted resistors: Example (from Gopalan)

* \( \Delta V_A \) for input 1111 1111 = 10.162 – 9.764 \( \approx \) 0.4 V which is larger than the resolution (0.039 V) of the DAC. This situation is not acceptable.

* The output voltage variation can be reduced by using resistors with a smaller tolerance. However, it is difficult to fabricate an IC with widely varying resistance values (from \( R \) to \( 2^{N-1}R \)) and each with a small enough tolerance.

\( \rightarrow \) use \( R – 2R \) ladder network instead.

M. B. Patil, IIT Bombay
Node $A_k$ is connected to $V_R$ if input bit $S_k$ is 1; else, it is connected to ground.
Node $A_k$ is connected to $V_R$ if input bit $S_k$ is 1; else, it is connected to ground.

The original network is equivalent to
R-2R ladder network: Thevenin resistance

\[ R_{\text{Th}} = R \]

M. B. Patil, IIT Bombay
R-2R ladder network: Thevenin resistance
R-2R ladder network: Thevenin resistance

\[ \text{Th} = R \]

M. B. Patil, IIT Bombay
R-2R ladder network: Thevenin resistance
R-2R ladder network: Thevenin resistance

\[ R_{Th} = R \]

M. B. Patil, IIT Bombay
R-2R ladder network: Thevenin resistance

\[ R_{Th} = R \]

M. B. Patil, IIT Bombay
R-2R ladder network: Thevenin resistance

\[ R_{\text{Th}} = R \]

M. B. Patil, IIT Bombay
R-2R ladder network: Thevenin resistance

\[ R_{Th} = R \]

M. B. Patil, IIT Bombay
R-2R ladder network: Thevenin resistance

\[ R_{Th} = R \]
R-2R ladder network:
\( V_{Th} \) for \( S_0 = 1 \)
R-2R ladder network:
\( V_{Th} \) for \( S_0 = 1 \)

\[ V_{Th} = V_R \]
R-2R ladder network:
$V_{Th}$ for $S_0 = 1$
R-2R ladder network: \( V_{Th} \) for \( S_0 = 1 \)
R-2R ladder network:
$V_{Th}$ for $S_0 = 1$
R-2R ladder network:
$V_{Th}$ for $S_0 = 1$

\[ V_{Th} = V_R \]
R-2R ladder network: 
$V_{Th}$ for $S_0 = 1$
R-2R ladder network:
$V_{Th}$ for $S_0 = 1$
R-2R ladder network: $V_{Th}$ for $S_0 = 1$

$V_{Th} = \frac{V_R}{16}$
R-2R ladder network: $V_{Th}$ for $S_1 = 1$
R-2R ladder network:
$V_{Th}$ for $S_1 = 1$
R-2R ladder network: $V_{Th}$ for $S_1 = 1$
R-2R ladder network: $V_{Th}$ for $S_1 = 1$
R-2R ladder network:
$V_{Th}$ for $S_1 = 1$
R-2R ladder network: $V_{Th}$ for $S_1 = 1$
R-2R ladder network:
$V_{Th}$ for $S_1 = 1$
R-2R ladder network:
$V_{Th}$ for $S_1 = 1$
R-2R ladder network:
$V_{Th}$ for $S_1 = 1$

$V_{Th} = \frac{V_R}{8}$
R-2R ladder network:
$V_{Th}$ for $S_2 = 1$
R-2R ladder network:
\( V_{Th} \) for \( S_2 = 1 \)
R-2R ladder network: $V_{Th}$ for $S_2 = 1$
R-2R ladder network:
$V_{Th}$ for $S_2 = 1$
R-2R ladder network:
$V_{Th}$ for $S_2 = 1$
R-2R ladder network:
$V_{Th}$ for $S_2 = 1$
R-2R ladder network:

\[ V_{Th} \text{ for } S_2 = 1 \]
R-2R ladder network:
\( V_{Th} \) for \( S_3 = 1 \)
R-2R ladder network: $V_{Th}$ for $S_3 = 1$
R-2R ladder network: 
$V_{Th}$ for $S_3 = 1$
R-2R ladder network: 
$V_{Th}$ for $S_3 = 1$

$V_{Th} = \frac{V_R}{2}$

M. B. Patil, IIT Bombay
R-2R ladder network: $R_{Th}$ and $V_{Th}$

We can use the R-2R ladder network and an op-amp to make up a DAC → next slide.
R-2R ladder network: $R_{Th}$ and $V_{Th}$

\[ R_{Th} = R. \]

We can use the R-2R ladder network and an op-amp to make up a DAC.
R-2R ladder network: $R_{Th}$ and $V_{Th}$

* $R_{Th} = R$

* $V_{Th} = V_{Th}^{(S0)} + V_{Th}^{(S1)} + V_{Th}^{(S2)} + V_{Th}^{(S3)}$

  \[ = \frac{V_R}{16} \left[ S_0 2^0 + S_1 2^1 + S_2 2^2 + S_3 2^3 \right]. \]
R-2R ladder network: $R_{Th}$ and $V_{Th}$

\[ R_{Th} = R. \]

\[ V_{Th} = V_{Th}^{(S_0)} + V_{Th}^{(S_1)} + V_{Th}^{(S_2)} + V_{Th}^{(S_3)} \]

\[ = \frac{V_R}{16} \left[ S_0 2^0 + S_1 2^1 + S_2 2^2 + S_3 2^3 \right]. \]

* We can use the R-2R ladder network and an op-amp to make up a DAC → next slide.
For an N-bit DAC,
\[ V_o = -R_f R_{Th} V_{Th} = -R_f R_{Th} V_R N - 1 \sum_{k=0}^{N-1} S_k R_k^2. \]
DAC with R-2R ladder

\[ V_o = -\frac{R_f}{R_{Th}} \cdot V_{Th} = -\frac{R_f}{R_{Th}} \cdot \frac{V_R}{16} \left[ S_0 \cdot 2^0 + S_1 \cdot 2^1 + S_2 \cdot 2^2 + S_3 \cdot 2^3 \right]. \]
DAC with R-2R ladder

\[ V_o = -\frac{R_f}{R_{Th}} V_{Th} = -\frac{R_f}{R_{Th}} \frac{V_R}{16} \left[ S_0 2^0 + S_1 2^1 + S_2 2^2 + S_3 2^3 \right] . \]

* For an N-bit DAC, \( V_o = -\frac{R_f}{R_{Th}} V_{Th} = -\frac{R_f}{R_{Th}} \frac{V_R}{2^N} \sum_{k=0}^{N-1} S_k 2^k . \)
DAC with R-2R ladder

\[ V_o = - \frac{R_f}{R_{Th}} V_{Th} = - \frac{R_f}{R_{Th}} \frac{V_R}{16} \left[ S_0 2^0 + S_1 2^1 + S_2 2^2 + S_3 2^3 \right]. \]

* For an N-bit DAC, \( V_o = - \frac{R_f}{R_{Th}} V_{Th} = - \frac{R_f}{R_{Th}} \frac{V_R}{2^N} \sum_{0}^{N-1} S_k 2^k. \)

* 6- to 20-bit DACs based on the R-2R ladder network are commercially available in monolithic form (single chip).

M. B. Patil, IIT Bombay
DAC with R-2R ladder

* \( V_o = -\frac{R_f}{R_{Th}} V_{Th} = -\frac{R_f}{R_{Th}} \frac{V_R}{16} \left[ S_0 2^0 + S_1 2^1 + S_2 2^2 + S_3 2^3 \right] \).

* For an N-bit DAC, \( V_o = -\frac{R_f}{R_{Th}} V_{Th} = -\frac{R_f}{R_{Th}} \frac{V_R}{2^N} \sum_{0}^{N-1} S_k 2^k \).

* 6- to 20-bit DACs based on the R-2R ladder network are commercially available in monolithic form (single chip).

* Bipolar, CMOS, or BiCMOS technology is used for these DACs.
DAC: home work

Combination of weighted−resistor and R–2R ladder networks

Find the value of $r$ for the circuit to work as a regular (i.e., binary to analog) DAC.

Find the value of $r$ for the circuit to work as a BCD to analog DAC.

M. B. Patil, IIT Bombay
Combination of weighted–resistor and R–2R ladder networks

* Find the value of $r$ for the circuit to work as a regular (i.e., binary to analog) DAC.
Combination of weighted−resistor and R−2R ladder networks

* Find the value of $r$ for the circuit to work as a regular (i.e., binary to analog) DAC.
* Find the value of $r$ for the circuit to work as a BCD to analog DAC.
When there is a change in the input binary number, the output $V_A$ takes a finite time to settle to the new value. The finite settling time arises because of stray capacitances and switching delays of the semiconductor devices used within the DAC chip. Example: 500 ns to 0.2% of full scale.
* When there is a change in the input binary number, the output $V_A$ takes a finite time to settle to the new value.
* When there is a change in the input binary number, the output $V_A$ takes a finite time to settle to the new value.

* The finite settling time arises because of stray capacitances and switching delays of the semiconductor devices used within the DAC chip.
* When there is a change in the input binary number, the output $V_A$ takes a finite time to settle to the new value.

* The finite settling time arises because of stray capacitances and switching delays of the semiconductor devices used within the DAC chip.

* Example: 500 ns to 0.2 % of full scale.
If the input $V_A$ is in the range $V_k R < V_A < V_{k+1} R$, the output is the binary number corresponding to the integer $k$. For example, for $V_A = V'_A$, the output is 100.

We may think of each voltage interval (corresponding to 000, 001, etc.) as a “bin.” In the above example, the input voltage $V'_A$ falls in the 100 bin; therefore, the output of the ADC would be 100.

Note that, for an N-bit ADC, there would be $2^N$ bins.
* If the input $V_A$ is in the range $V_R^k < V_A < V_R^{k+1}$, the output is the binary number corresponding to the integer $k$. For example, for $V_A = V_A'$, the output is 100.
* If the input $V_A$ is in the range $V^k_R < V_A < V^{k+1}_R$, the output is the binary number corresponding to the integer $k$. For example, for $V_A = V'_R$, the output is 100.

* We may think of each voltage interval (corresponding to 000, 001, etc.) as a “bin.” In the above example, the input voltage $V'_A$ falls in the 100 bin; therefore, the output of the ADC would be 100.
* If the input $V_A$ is in the range $V_R^k < V_A < V_R^{k+1}$, the output is the binary number corresponding to the integer $k$. For example, for $V_A = V'_A$, the output is 100.

* We may think of each voltage interval (corresponding to 000, 001, etc.) as a “bin.” In the above example, the input voltage $V'_A$ falls in the 100 bin; therefore, the output of the ADC would be 100.

* Note that, for an $N$-bit ADC, there would be $2^N$ bins.
The basic idea behind an ADC is simple:
- Generate reference voltages $V_1^R, V_2^R$, etc.
- Compare the input $V_A$ with each of $V_i^R$ to figure out which bin it belongs to.
- If $V_A$ belongs to bin $k$ (i.e., $V_k^R < V_A < V_{k+1}^R$), convert $k$ to the binary format.

A "parallel" ADC does exactly that. 🔄
The basic idea behind an ADC is simple:

- Generate reference voltages $V_1$, $V_2$, etc.
- Compare the input $V_A$ with each of $V_i$ to figure out which bin it belongs to.
- If $V_A$ belongs to bin $k$ (i.e., $V_k < V_A < V_k + 1$), convert $k$ to the binary format.

A “parallel” ADC does exactly that → next slide.
The basic idea behind an ADC is simple:

- Generate reference voltages $V_R^1$, $V_R^2$, etc.
The basic idea behind an ADC is simple:

- Generate reference voltages $V_R^1$, $V_R^2$, etc.
- Compare the input $V_A$ with each of $V_R^i$ to figure out which bin it belongs to.
The basic idea behind an ADC is simple:

- Generate reference voltages $V_R^1$, $V_R^2$, etc.
- Compare the input $V_A$ with each of $V_R^i$ to figure out which bin it belongs to.
- If $V_A$ belongs to bin $k$ (i.e., $V_R^k < V_A < V_R^{k+1}$), convert $k$ to the binary format.
The basic idea behind an ADC is simple:

- Generate reference voltages $V^1_R$, $V^2_R$, etc.
- Compare the input $V_A$ with each of $V^i_R$ to figure out which bin it belongs to.
- If $V_A$ belongs to bin $k$ (i.e., $V^k_R < V_A < V^{k+1}_R$), convert $k$ to the binary format.

A “parallel” ADC does exactly that → next slide.
3-bit parallel (flash) ADC

M. B. Patil, IIT Bombay
Practical difficulty: As the input changes, the comparator outputs \((C_0, C_1, \text{etc.})\) may not settle to their new values at the same time. → ADC output will depend on when we sample it.

Add D flip-flops. Allow sufficient time (between the change in \(V_A\) and the active clock edge) so that the comparator outputs have already settled to their new values before they get latched in.

M. B. Patil, IIT Bombay
3-bit parallel (flash) ADC

* Practical difficulty: As the input changes, the comparator outputs ($C_0$, $C_1$, etc.) may not settle to their new values at the same time. → ADC output will depend on when we sample it.
3-bit parallel (flash) ADC

* Practical difficulty: As the input changes, the comparator outputs \( C_0, C_1, \text{etc.} \) may not settle to their new values at the same time. \( \rightarrow \) ADC output will depend on when we sample it.

* Add D flip-flops. Allow sufficient time (between the change in \( V_A \) and the active clock edge) so that the comparator outputs have already settled to their new values before they get latched in.
Practical difficulty: As the input changes, the comparator outputs ($C_0$, $C_1$, etc.) may not settle to their new values at the same time. → ADC output will depend on when we sample it.

Add D flip-flops. Allow sufficient time (between the change in $V_A$ and the active clock edge) so that the comparator outputs have already settled to their new values before they get latched in.
In the parallel (flash) ADC, the conversion gets done “in parallel,” since all comparators operate on the same input voltage.

Conversion time is governed only by the comparator response time → fast conversion (hence the name “flash” converter).

2N comparators are required for N-bit ADC → generally limited to 8 bits.
* In the parallel (flash) ADC, the conversion gets done “in parallel,” since all comparators operate on the same input voltage.

* Conversion time is governed only by the comparator response time → fast conversion (hence the name “flash” converter).
In the parallel (flash) ADC, the conversion gets done “in parallel,” since all comparators operate on the same input voltage.

Conversion time is governed only by the comparator response time → fast conversion (hence the name “flash” converter).

Flash ADCs to handle 500 million analog samples per second are commercially available.
* In the parallel (flash) ADC, the conversion gets done “in parallel,” since all comparators operate on the same input voltage.

* Conversion time is governed only by the comparator response time → fast conversion (hence the name “flash” converter).

* Flash ADCs to handle 500 million analog samples per second are commercially available.

* $2^N$ comparators are required for N-bit ADC → generally limited to 8 bits.
An ADC typically operates on a “sampled” input signal ($V_s(t)$ in the figure) which is derived from the continuously varying input signal ($V_a(t)$ in the figure) with a “sample-and-hold” (S/H) circuit. The S/H circuit samples the input signal $V_a(t)$ at uniform intervals of duration $T_c$, the clock period. When the clock goes high, switch S (e.g., a FET or a CMOS pass gate) is closed, and the capacitor C gets charged to the signal voltage at that time. When the clock goes low, switch S is turned off, and C holds the voltage constant, as desired. Op-amp buffers can be used to minimise loading effects.
An ADC typically operates on a “sampled” input signal ($V_s(t)$ in the figure) which is derived from the continuously varying input signal ($V_a(t)$ in the figure) with a “sample-and-hold” (S/H) circuit.
An ADC typically operates on a “sampled” input signal \( V_s(t) \) in the figure) which is derived from the continuously varying input signal \( V_a(t) \) in the figure) with a “sample-and-hold” (S/H) circuit.

- The S/H circuit samples the input signal \( V_a(t) \) at uniform intervals of duration \( T_c \), the clock period.
An ADC typically operates on a “sampled” input signal \(V_s(t)\) in the figure) which is derived from the continuously varying input signal \(V_a(t)\) in the figure) with a “sample-and-hold” (S/H) circuit.

* The S/H circuit samples the input signal \(V_a(t)\) at uniform intervals of duration \(T_c\), the clock period.

* When the clock goes high, switch \(S\) (e.g., a FET or a CMOS pass gate) is closed, and the capacitor \(C\) gets charged to the signal voltage at that time. When the clock goes low, switch \(S\) is turned off, and \(C\) holds the voltage constant, as desired.
An ADC typically operates on a “sampled” input signal ($V_s(t)$ in the figure) which is derived from the continuously varying input signal ($V_a(t)$ in the figure) with a “sample-and-hold” (S/H) circuit.

The S/H circuit samples the input signal $V_a(t)$ at uniform intervals of duration $T_c$, the clock period.

When the clock goes high, switch S (e.g., a FET or a CMOS pass gate) is closed, and the capacitor C gets charged to the signal voltage at that time. When the clock goes low, switch S is turned off, and C holds the voltage constant, as desired.

Op-amp buffers can be used to minimise loading effects.
Suppose we have a 4-bit DAC. We can use it to perform A-to-D conversion by successively setting the four bits as follows.

- Start with $D_3 D_2 D_1 D_0 = 0000$, $I = 3$.
- Set $D \left[ I \right] = 1$ (keep other bits unchanged).
- If $V_{DAC_o} > V_A$ (i.e., $C = 0$), set $D \left[ I \right] = 0$; else, keep $D \left[ I \right] = 1$.
- $I \leftarrow I - 1$; go to step 1.

At the end of four steps, the digital output is given by $D_3 D_2 D_1 D_0$.
Suppose we have a 4-bit DAC. We can use it to perform A-to-D conversion by successively setting the four bits as follows.

1. Start with $D_3D_2D_1D_0 = 0000$, $I = 3$.
2. Set $D[I] = 1$ (keep other bits unchanged).
3. If $V_{DACo} > V_A$ (i.e., $C = 0$), set $D[I] = 0$; else, keep $D[I] = 1$.
4. $I \leftarrow I - 1$; go to step 1.

At the end of four steps, the digital output is given by $D_3D_2D_1D_0$. Example $\rightarrow$ next slide.
Suppose we have a 4-bit DAC. We can use it to perform A-to-D conversion by successively setting the four bits as follows.

- Start with $D_3 D_2 D_1 D_0 = 0000$, $I = 3$. 

At the end of four steps, the digital output is given by $D_3 D_2 D_1 D_0$. 

Example → next slide.
Successive Approximation ADC

Suppose we have a 4-bit DAC. We can use it to perform A-to-D conversion by successively setting the four bits as follows.

- Start with $D_3D_2D_1D_0 = 0000$, $I = 3$.
- Set $D[I] = 1$ (keep other bits unchanged).
Suppose we have a 4-bit DAC. We can use it to perform A-to-D conversion by successively setting the four bits as follows.

- Start with $D_3D_2D_1D_0 = 0000$, $I = 3$.
- Set $D[I] = 1$ (keep other bits unchanged).
- If $V^{DAC}_o > V_A$ (i.e., $C = 0$), set $D[I] = 0$; else, keep $D[I] = 1$. 

* Example → next slide.
Suppose we have a 4-bit DAC. We can use it to perform A-to-D conversion by successively setting the four bits as follows.

- Start with $D_3 D_2 D_1 D_0 = 0000$, $I = 3$.
- Set $D[I] = 1$ (keep other bits unchanged).
- If $V_o^{DAC} > V_A$ (i.e., $C = 0$), set $D[I] = 0$; else, keep $D[I] = 1$.
- $I \leftarrow I - 1$; go to step 1.
Suppose we have a 4-bit DAC. We can use it to perform A-to-D conversion by successively setting the four bits as follows.

- Start with $D_3D_2D_1D_0 = 0000$, $I = 3$.
- Set $D[I] = 1$ (keep other bits unchanged).
- If $V_{DAC}^o > V_A$ (i.e., $C = 0$), set $D[I] = 0$; else, keep $D[I] = 1$.
- $I \leftarrow I - 1$; go to step 1.

* At the end of four steps, the digital output is given by $D_3D_2D_1D_0$.

Example → next slide.
Successive Approximation ADC

(Note: $k \propto V_R$)

At the end of the 5th step, we know that the input voltage corresponds to $10110$. For the digital representation to be accurate up to $\pm \frac{1}{2}$ LSB, $\Delta V$ corresponding to $\frac{1}{2}$ LSB is added to $V_A$ (see [Taub]).
At the end of the 5th step, we know that the input voltage corresponds to 10110.
Successive Approximation ADC

```
D4 = 1  D3 = 1  D4 = 1  D4 = 1  D4 = 1
D3 = 0  D3 = 1  D3 = 0  D3 = 0  D3 = 0
D2 = 0  D2 = 0  D2 = 1  D2 = 1  D2 = 1
D1 = 0  D1 = 0  D1 = 0  D1 = 1  D1 = 1
D0 = 0  D0 = 0  D0 = 0  D0 = 0  D0 = 1
C = 1   C = 0   C = 1   C = 1   C = 0
→ reset D3  \rightarrow \text{reset } D0
```

(Note: \( k \propto V_R \))

* At the end of the 5th step, we know that the input voltage corresponds to 10110.
* For the digital representation to be accurate up to \( \pm \frac{1}{2} \) LSB, \( \Delta V \) corresponding to \( \frac{1}{2} \) LSB is added to \( V_A \) (see [Taub]).
Successive Approximation ADC

Each step (setting SAR bits, comparison of $V_A$ and $V_{DAC}$) is performed in one clock cycle → conversion time is $N$ cycles, irrespective of the input voltage value $V_A$.

S. A. ADCs with built-in or external S/H (sample-and-hold) are available for 8- to 16-bit resolution and conversion times of a few $\mu$sec to tens of $\mu$sec.

Useful for medium-speed applications such as speech transmission with PCM.

M. B. Patil, IIT Bombay
Successive Approximation ADC

* Each step (setting SAR bits, comparison of $V_A$ and $V_{DAC}^o$) is performed in one clock cycle → conversion time is $N$ cycles, irrespective of the input voltage value $V_A$. 

M. B. Patil, IIT Bombay
Each step (setting SAR bits, comparison of $V_A$ and $V_{DAC}^o$) is performed in one clock cycle → conversion time is $N$ cycles, irrespective of the input voltage value $V_A$.

* S. A. ADCs with built-in or external S/H (sample-and-hold) are available for 8- to 16-bit resolution and conversion times of a few µsec to tens of µsec.
Successive Approximation ADC

Each step (setting SAR bits, comparison of $V_A$ and $V_{oDAC}^D$) is performed in one clock cycle → conversion time is $N$ cycles, irrespective of the input voltage value $V_A$.

S. A. ADCs with built-in or external S/H (sample-and-hold) are available for 8- to 16-bit resolution and conversion times of a few $\mu$sec to tens of $\mu$sec.

Useful for medium-speed applications such as speech transmission with PCM.

M. B. Patil, IIT Bombay
The "start conversion" signal clears the counter; counting begins, and $V_{\text{DAC}}$ increases with each clock cycle. When $V_{\text{DAC}}$ exceeds $V_A$, $C$ becomes 0, and counting stops.

Simple scheme, but (a) conversion time depends on $V_A$, (b) slow (takes $(2^N - 1)$ clock cycles in the worst case) → tracking ADC

M. B. Patil, IIT Bombay
* The “start conversion” signal clears the counter; counting begins, and $V_{DAC}^o$ increases with each clock cycle.
* The “start conversion” signal clears the counter; counting begins, and $V_{DAC}^o$ increases with each clock cycle.

* When $V_{DAC}^o$ exceeds $V_A$, $C$ becomes 0, and counting stops.
Counting ADC (digital-ramp ADC)

* The “start conversion” signal clears the counter; counting begins, and $V_{DAC}^o$ increases with each clock cycle.
* When $V_{DAC}^o$ exceeds $V_A$, $C$ becomes 0, and counting stops.
* Simple scheme, but (a) conversion time depends on $V_A$, (b) slow (takes $(2^N - 1)$ clock cycles in the worst case) → tracking ADC
The counter counts up if $V_{DAC_o} < V_A$; else, it counts down.

If $V_A$ changes, the counter does not need to start from 000···0, so the conversion time is less than that required by a counting ADC.

* used in low-cost, low-speed applications, e.g., measuring output from a temperature sensor or a strain gauge

M. B. Patil, IIT Bombay
* The counter counts up if $V_{o}^{DAC} < V_A$; else, it counts down.

---

M. B. Patil, IIT Bombay
* The counter counts up if $V_{DAC}^o < V_A$; else, it counts down.
* If $V_A$ changes, the counter does not need to start from 000···0, so the conversion time is less than that required by a counting ADC.
The counter counts up if $V^D_{DAC} < V_A$; else, it counts down.

If $V_A$ changes, the counter does not need to start from 000· · · 0, so the conversion time is less than that required by a counting ADC.

* used in low-cost, low-speed applications, e.g., measuring output from a temperature sensor or a strain gauge
**Dual-slope ADC**

\[
V^o = -\frac{1}{RC} \int V_i \, dt
\]

\[
\text{slope} = -\frac{V_A}{RC}
\]

\[
\text{slope} = -\frac{V_R}{RC}
\]

\[
0 \quad T_1 \quad T_2 \quad t
\]

\[
-V_1 \quad V_A \quad 0
\]

- **t**: 0: reset integrator output $V^o$ to 0 V by closing S momentarily.
- **t**: $T_1$, integrator output reaches $-V_1 = -V_A T_1 / RC$.
- **Now apply a reference voltage $V_R$ (assumed to be negative, with $|V_R| > V_A$), and integrate until $V^o$ reaches 0 V.**
- **Since** $V_1 = V_A T_1 / RC = |V_R| T_2 / RC$, we have $T_2 = T_1 V_A / |V_R| \to T_2$ gives a measure of $V_A$.
- **In the dual-slope ADC**, a counter output – which is proportional to $T_2$ – provides the desired digital output.
Dual-slope ADC

* $t = 0$: reset integrator output $V_o$ to 0 V by closing $S$ momentarily.
Dual-slope ADC

\[ V_o = -\frac{1}{RC} \int V_i \, dt \]

* \( t = 0 \): reset integrator output \( V_o \) to 0 V by closing \( S \) momentarily.
* Integrate \( V_A \) (voltage to be converted to digital format, assumed to be positive) for a fixed interval \( T_1 \).
* $t = 0$: reset integrator output $V_o$ to 0 V by closing $S$ momentarily.
* Integrate $V_A$ (voltage to be converted to digital format, assumed to be positive) for a fixed interval $T_1$.
* At $t = T_1$, integrator output reaches $-V_1 = -V_A \frac{T_1}{RC}$.
Dual-slope ADC

\[ V_o = - \frac{1}{RC} \int V_i \, dt \]

* \( t = 0 \): reset integrator output \( V_o \) to 0 V by closing \( S \) momentarily.
* Integrate \( V_A \) (voltage to be converted to digital format, assumed to be positive) for a fixed interval \( T_1 \).
* At \( t = T_1 \), integrator output reaches \( -V_1 = -V_A \frac{T_1}{RC} \).
* Now apply a reference voltage \( V_R \) (assumed to be negative, with \( |V_R| > V_A \)), and integrate until \( V_o \) reaches 0 V.
Dual-slope ADC

\[ V_o = -\frac{1}{RC} \int V_i \, dt \]

\[ \text{slope} = -\frac{V_A}{RC} \]

\[ \text{slope} = -\frac{V_R}{RC} \]

\[ T_2 = T_1 \frac{V_A}{|V_R|} \rightarrow T_2 \text{ gives a measure of } V_A. \]

* \( t = 0 \): reset integrator output \( V_o \) to 0 V by closing \( S \) momentarily.
* Integrate \( V_A \) (voltage to be converted to digital format, assumed to be positive) for a fixed interval \( T_1 \).
* At \( t = T_1 \), integrator output reaches \( -V_1 = -V_A \frac{T_1}{RC} \).
* Now apply a reference voltage \( V_R \) (assumed to be negative, with \( |V_R| > V_A \)), and integrate until \( V_o \) reaches 0 V.

In the dual-slope ADC, a counter output – which is proportional to \( T_2 \) – provides the desired digital output.
In the dual-slope ADC, a counter output – which is proportional to \( T_2 \) – provides the desired digital output.

\[ V_o = -\frac{1}{RC} \int V_i \, dt \]

\[ \text{slope} = -\frac{V_A}{RC} \]

\[ \text{slope} = -\frac{V_R}{RC} \]

* \( t = 0 \): reset integrator output \( V_o \) to 0 V by closing \( S \) momentarily.

* Integrate \( V_A \) (voltage to be converted to digital format, assumed to be positive) for a fixed interval \( T_1 \).

* At \( t = T_1 \), integrator output reaches \( -V_1 = -V_A \frac{T_1}{RC} \).

* Now apply a reference voltage \( V_R \) (assumed to be negative, with \( |V_R| > V_A \)), and integrate until \( V_o \) reaches 0 V.

* Since \( V_1 = V_A \frac{T_1}{RC} = |V_R| \frac{T_2}{RC} \), we have \( T_2 = T_1 \frac{V_A}{|V_R|} \rightarrow T_2 \) gives a measure of \( V_A \).

M. B. Patil, IIT Bombay
Dual-slope ADC

- $V_A$: input voltage
- $V_R$: reference voltage
- $R$: resistor
- $C$: capacitor
- $V_o$: comparator output
- $T_1 = 2^N T_c$: time for the first ramp
- $T_2$: time for the second ramp
- $T_c$: clock period
- $V_{o}$: output voltage
- $T_c$: clock period
- $T_1 = 2^N T_c$: time for the first ramp
- $T_2$: time for the second ramp
- $T_c$: clock period
- $V_{o}$: output voltage

M. B. Patil, IIT Bombay
Dual-slope ADC

* Start: counter reset to 000····0, SPDT in position A.

\[ T_1 = 2^N T_c \]

\[ T_2 \]

\[ \text{slope} = -\frac{V_A}{RC} \]

\[ \text{slope} = -\frac{V_R}{RC} \]
Start: counter reset to 000··0, SPDT in position A.

Counter counts up to $2^N$ at which point the overflow flag becomes 1, and SPDT switches to position B → $T_1 = 2^N T_c$ where $T_c$ is the clock period.
* Start: counter reset to 000⋯0, SPDT in position A.
* Counter counts up to $2^N$ at which point the overflow flag becomes 1, and SPDT switches to position B $\rightarrow T_1 = 2^N T_c$ where $T_c$ is the clock period.
* The counter starts counting again from 000⋯0, and stops counting when $V_o$ crosses 0 V. The counter output gives $T_2$ in binary format.