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.
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 $S_3 S_2 S_1 S_0$, 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, $V_A = K \sum_{0}^{N-1} S_k 2^k$. 
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, $V_A = K \sum_{k=0}^{N-1} S_k 2^k$.

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

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

$V_A = -R_f I = -V_R R_f \sum_{k=0}^{N-1} S_k \times 2^k$.

Using $R_k = 2^{N-1} R / 2^k$, we get

$V_A = -V_R R_f 2^{N-1} \sum_{k=0}^{N-1} S_k$.

M. B. Patil, IIT Bombay
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.
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. → $V(A_k) = S_k \times V_R$. 

M. B. Patil, IIT Bombay
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,
  $I_k = \frac{V(A_k) - 0}{R_k} = \frac{S_k V_R}{R_k}$.
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,
  $I_k = \frac{V(A_k) - 0}{R_k} = \frac{S_k \times V_R}{R_k}$.

* Using $R_k = 2^{N-1} R/2^k$, we get $I = \frac{V_R}{2^{N-1} R} \sum_{0}^{N-1} S_k \times 2^k$ (for $N = 4$ here).
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. 
  \[ V(A_k) = S_k \times V_R \]

* Since the inverting terminal of the Op Amp is at virtual ground,
  \[ I_k = \frac{V(A_k) - 0}{R_k} = \frac{S_k \times V_R}{R_k} \]

* Using $R_k = 2^{N-1} R/2^k$, we get
  \[ I = \frac{V_R}{2^{N-1}R} \sum_{0}^{N-1} S_k \times 2^k \] (N = 4 here).

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

$10\, \text{mA} = V_R \frac{R}{2} + V_R \frac{1}{2^2} R + \cdots + V_R \frac{1}{2^7} R = \frac{1}{2^7} V_R (2^8 - 1) = \frac{255}{128} V_R$

→ $R_{\text{min}} = \frac{5\, \text{V}}{10\, \text{mA} \times \frac{255}{128}} = 996\, \Omega$. 

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?
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 $V_R$. 

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

* 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 $V_R$.

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

\[= \frac{1}{2^7} \frac{V_R}{R} \left(2^8 - 1\right) = \frac{255}{128} \frac{V_R}{R}\]
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 mA?

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

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

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

$$= \frac{1}{2^7} \frac{V_R}{R} \left(2^8 - 1\right) = \frac{255}{128} \frac{V_R}{R}$$

→ $R_{\text{min}} = \frac{5\, \text{V}}{10\, \text{mA}} \times \frac{255}{128} = 996\, \Omega$. 

M. B. Patil, IIT Bombay
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)?

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

$$\Delta V_A = V_R 2^N - \frac{V_R}{R_f} = \frac{5 \text{V}}{2^8} - \frac{5 \text{V}}{1} = \frac{5 \text{V}}{128} = 0.0391 \text{V}.$$
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)?
* 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)?

\[
V_A = -V_R \frac{R_f}{2^{N-1}R} \left[ S_72^7 + \cdots + S_12^1 + S_02^0 \right]
\]
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)?

$$V_A = -V_R \frac{2^N - 1}{2^N} \left( 2^0 + 2^1 + \cdots + 2^7 \right) = 5 \frac{128}{128} \left( 2^8 - 1 \right) = 5 \times 255 = 996 \text{ V}.$$
DAC using binary-weighted resistors: Example (from Gopalan)

* What is the maximum output voltage (in magnitude)?

\[ V_A = -V_R - \left( \frac{R_7}{R_f} \right) \left( \frac{R_1}{2^6 R} \right) + \cdots + \left( \frac{R_0}{2^7 R} \right) \]

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

\[ |V_A|_{\text{max}} = 5 \times 2^{128} \times 1 \times \left( 2^{8} - 1 \right) = 5 \times 2^{5} = 9.961 \text{ V} \]
* What is the maximum output voltage (in magnitude)?

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

\[ |V_A|_{\text{max}} = \frac{5}{128} \times 1 \times \left[ 2^0 + 2^1 + \cdots + 2^7 \right] = \frac{5}{128} \times \left( 2^8 - 1 \right) = 5 \times \frac{255}{128} = 9.961 \text{ V}. \]
Find the output voltage corresponding to the input 1010 1101.

\[ V_A = -V_R \left( \frac{2^7 R_0}{R_f} \right) \]

\[ = -5 \times \frac{1}{128} \times \left( 2^7 + 2^5 + 2^3 + 2^2 + 2^0 \right) \]

\[ = -6.758 \text{ V} \]
* Find the output voltage corresponding to the input 1010 1101.
DAC using binary-weighted resistors: Example (from Gopalan)

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

* Find the output voltage corresponding to the input 1010 1101.

M. B. Patil, IIT Bombay
* Find the output voltage corresponding to the input 1010 1101.

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

\[ = -\frac{5}{128} \times 1 \times \left[ 2^7 + 2^5 + 2^3 + 2^2 + 2^0 \right] = -5 \times \frac{173}{128} = -6.758 \text{ V} \]
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 $1111\,1111$?

$|V_A|_{\text{max}} = V_R \times \frac{255}{128} \times R_f R_{\text{max}} = 5 \times \frac{255}{128} \times 1.01 \times 0.99 = 10.162\,\text{V}$

Similarly, $|V_A|_{\text{min}} = 5 \times \frac{255}{128} \times 0.99 \times 1.01 = 9.764\,\text{V}$.

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?
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 \(I_0, I_1, \text{etc.}\) 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.)
* 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.)

\[
\rightarrow |V_A|_{11111111}^{\text{max}} = V_R \times \frac{255}{128} \times \frac{R_f^{\text{max}}}{R} = 5 \times \frac{255}{128} \times \frac{1.01}{0.99} = 10.162 \, \text{V}.
\]
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.)

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

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

\[ V_R \]
\[ \begin{align*}
I_7 & = R_7 = R \\
I_1 & = R_1 = 2^6 R \\
I_0 & = R_0 = 2^7 R
\end{align*} \]

\[ V_A \]

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

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.

\[ \rightarrow \text{ use } R - 2R \text{ ladder network instead.} \]
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.
DAC using binary-weighted resistors: Example (from Gopalan)

* $\Delta V_A$ 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 ≈ 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. → use $R - 2R$ ladder network instead.
Node $A_k$ is connected to $V_R$ if input bit $S_k$ is 1; else, it is connected to ground.
R-2R ladder network

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

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

\[ \text{Th} = R \]
R-2R ladder network: Thevenin resistance

\[ R_{Th} = R \]

M. B. Patil, IIT Bombay
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

\[ \text{Th} = R \]

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

\[ R_{Th} = R \]
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 $S_0 = 1$
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 $S_0 = 1$
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: $V_{Th}$ for $S_2 = 1$

\[ V_{Th} = \frac{V_R}{4} \]
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 $S_3 = 1$

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

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

* $R_{Th} = R$.
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,

\[ V_o = -R_f R_{Th} V_{Th} = -R_f R_{Th} V_R^{2N-1} \sum_{k=0}^{N} S_k 2^k. \]

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.
For an N-bit DAC,

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

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

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

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

* 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.
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.
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.

M. B. Patil, IIT Bombay
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.
ADC: introduction

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.

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 $V_A = V'_A$, 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.
* 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.
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 $V_R, V_{2R}, etc.$
- 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.
The basic idea behind an ADC is simple:

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

- Generate reference voltages $V_{R1}, V_{R2}$, etc.
- Compare the input $V_A$ with each of $V_{Ri}$ to figure out which bin it belongs to.
- If $V_A$ belongs to bin $k$ (i.e., $V_{Rk} < V_A < V_{Rk+1}$), convert $k$ to the binary format.
The basic idea behind an ADC is simple:

- Generate reference voltages $V_{R1}, V_{R2},$ etc.
- Compare the input $V_A$ with each of $V_{Ri}$ to figure out which bin it belongs to.
- If $V_A$ belongs to bin $k$ (i.e., $V_{Rk} < V_A < V_{R(k+1)}$), convert $k$ to the binary format.

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

3-bit ADC

analog input

V_A

digital output

D_2

D_1

D_0

ground

3-bit ADC

V_R

V_{max}

V_7

V_6

V_5

V_4

V_3

V_2

V_1

V_R

V'_A

111

110

101

100

011

010

001

000

LOGIC

D_2

D_1

D_0

V_A

V_R

R

R

R

R
3-bit parallel (flash) ADC

Practical difficulty: As the input changes, the comparator outputs (C₀, C₁, 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.
* 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.
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.
* 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.
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 $D_3 D_2 D_1 D_0 = 0000$, $I = 3$.
- 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 $D_3 D_2 D_1 D_0$. Example → next slide.

M. B. Patil, IIT Bombay
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 $D[I] = 1$.
* $I ← I - 1$; go to step 1.

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

Example → next slide.
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 $D_3D_2D_1D_0 = 0000$, $I = 3$. 

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 $D_3D_2D_1D_0 = 0000$, $I = 3$.
- 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 $V_o^{DAC} > V_A$ (i.e., $C = 0$), set $D[I] = 0$; else, keep $D[I] = 1$. 

\* At the end of four steps, the digital output is given by $D_3D_2D_1D_0$. 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 $D_3D_2D_1D_0 = 0000$, $I = 3$.
- 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.
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 \( D_3 D_2 D_1 D_0 = 0000 \), \( I = 3 \).
- 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.

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

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

![Diagram of 5-bit DAC]

\[ V_{DAC}^o \]

\[ V_R \]

\[ V_A \]

\[ C \]

(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 ±1/2 LSB, \( \Delta V \) corresponding to 1/2 LSB is added to \( V_A \) (see [Taub]).

M. B. Patil, IIT Bombay
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]).
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 $\mu$s to tens of $\mu$s.

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

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 $V_A$. 
* 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 $\mu$sec to tens of $\mu$sec.
Each step (setting SAR bits, comparison of $V_A$ and $V_o^{DAC}$) 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.
The "start conversion" signal clears the counter; counting begins, and $V_{DAC}$ increases with each clock cycle.

When $V_{DAC}$ exceeds $V_A$, $C$ becomes 0, and counting stops.

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

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

When $V_o^{DAC}$ exceeds $V_A$, $C$ becomes 0, and counting stops.
The “start conversion” signal clears the counter; counting begins, and $V_{DAC}^0$ increases with each clock cycle.

When $V_{DAC}^0$ exceeds $V_A$, $C$ becomes 0, and counting stops.

Simple scheme, but (a) conversion time depends on $V_A$, (b) slow (takes $2^N$ clock cycles in the worst case) → tracking ADC (next slide)
The counter counts up if $V_{DAC} < V_A$; else, it counts down.

If $V_A$ changes, the counter does not need to start from $000\cdots0$, 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.
* The counter counts up if $V_o^{DAC} < V_A$; else, it counts down.
The counter counts up if $V_o^{DAC} < V_A$; 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.
* The counter counts up if $V_{o}^{DAC} < V_A$; 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

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

- **S**: Start
- **C**: Capacitor
- **R**: Resistor
- **V_A**: Voltage to be converted
- **V_R**: Reference voltage
- **V_i**: Input voltage

1. **Reset Integrator**: Reset the integrator to 0 V by closing switch S momentarily.
2. **Integrate**: Integrate \( V_A \) (voltage to be converted to digital format, assumed to be positive) for a fixed interval \( T_1 \).
3. **Output**: At \( t = T_1 \), the integrator output reaches \( -V_1 = -V_A T_1/RC \).
4. **Reference Voltage**: Apply a reference voltage \( V_R \) (assumed to be negative, with \( |V_R| > V_A \)), and integrate until \( V_o \) reaches 0 V.
5. **Equation**: Since \( V_1 = V_A T_1/RC = |V_R| T_2/RC \), we have \( T_2 = T_1 V_A/|V_R| \rightarrow T_2 \) gives a measure of \( V_A \).
6. **Digital Output**: In the dual-slope ADC, a counter output – which is proportional to \( T_2 \) – provides the desired digital output.

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

\[
\begin{align*}
V_o &= \frac{-1}{RC} \int V_i \, dt \\
\text{slope} &= \frac{-V_A}{RC} \\
\text{slope} &= \frac{-V_R}{RC}
\end{align*}
\]

* \( t = 0 \): 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

* $t = 0$: 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.

* $t = 0$: 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 $V_A$. 

\[ V_o = -\frac{1}{RC} \int V_i \, dt \]
* $t = 0$: 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.

* Since $V_1 = V_A \frac{T_1}{RC} = |V_R| \frac{T_2}{RC}$, we have $T_2 = T_1 \frac{V_A}{|V_R|} \to T_2$ 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

- **SPDT**
- **V_A, V_R**
- **R**
- **C**
- **N-bit Counter**
- **Comparator**
- **Integrator**
- **Overflow**
- **Clock**
- **Digital Output**

Mathematical Expressions:

- \( \text{slope} = - \frac{V_A}{RC} \)
- \( \text{slope} = - \frac{V_R}{RC} \)
- \( T_1 = 2^N T_c \)
- \( T_2 = T_1 \)

Graph:

- Time axis (t)
- Voltage axis (V)
- Line slopes indicated

Equations:

- \( V_0 = \frac{V_R}{2^N} \)
- \( T_2 = 2^N T_c \)

Diagram Notes:

- Circuit includes SPDT switch, integrator, comparator, clock, and overflow.
- Time period \( T_c \)
- Digital output

M. B. Patil, IIT Bombay
* Start: counter reset to 000· · · 0, SPDT in position A.
Dual-slope ADC

* 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.
Dual-slope ADC

* 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.

* 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.