Lecture 15: Analog to Digital Converters (ADCs) and Digital to Analog Converters (DACs)

18-100: Introduction to ECE

SPRING 2025

Electrical & Computer ENGINEERING

Carnegie Mellon University

Agenda

- Analog and Digital Worlds – The Problem with Pizza
- Analog to Digital Converters (ADCs)
- Digital to Analog Converters (DACs)

Digital Inputs and Outputs

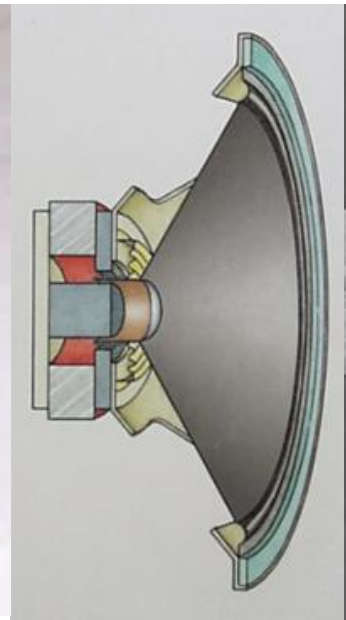
- Digital inputs and outputs (I/O) represent YES/NO or TRUE/FALSE information

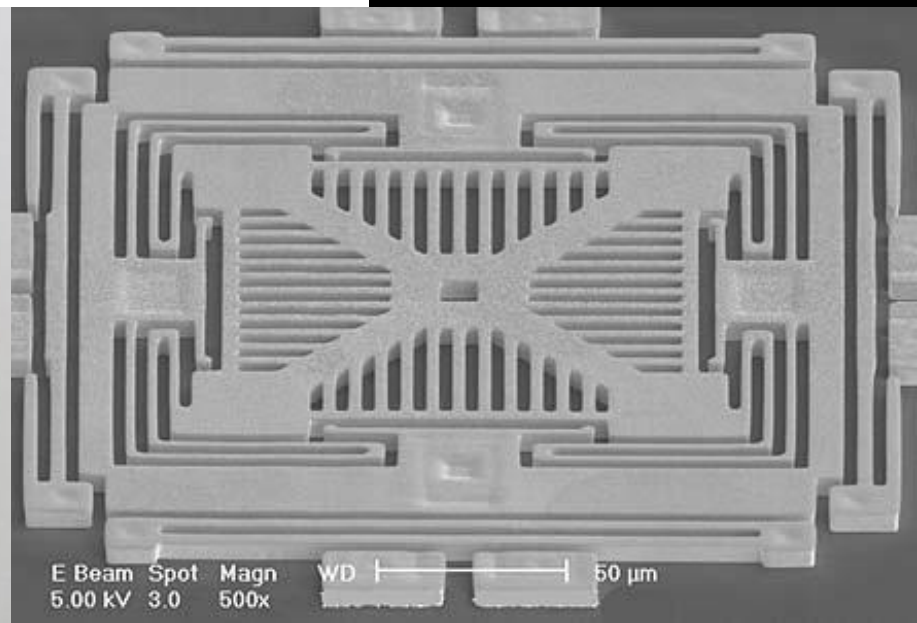
- Examples: The light is on.
  A button is pushed.
  Turn on a bell.
  Turn off a light emitting diode.

We Live in an Analog World

- Everything in the physical world is an analog signal: Sound, light, temperature, pressure….

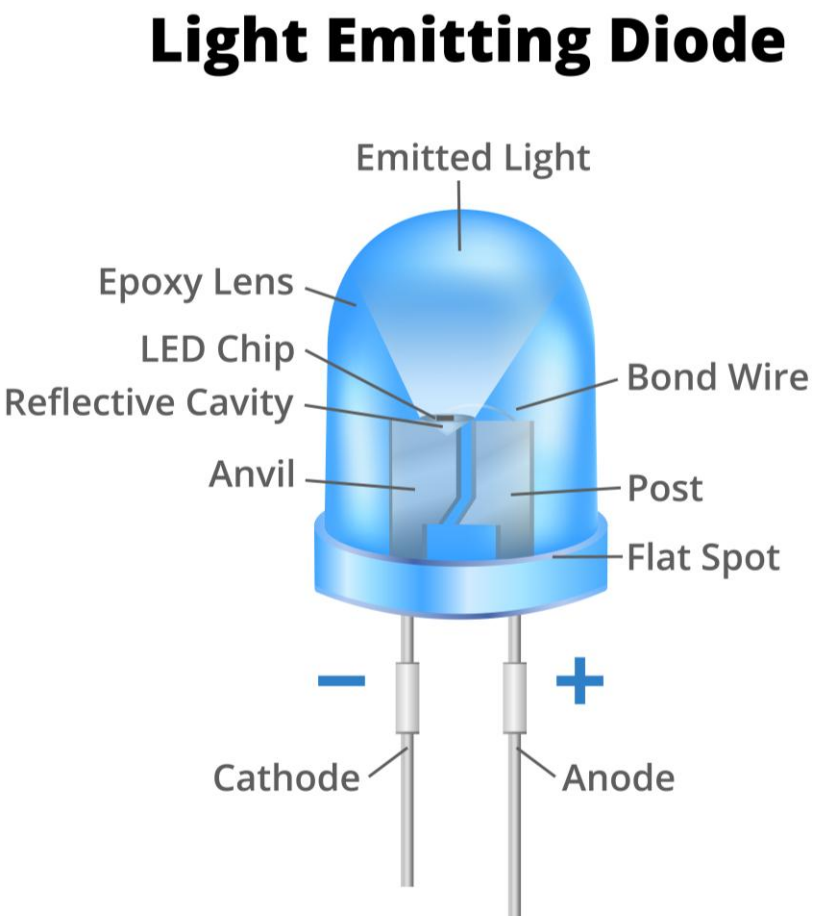
- Need to convert “real world” information into electrical signals
- Transducers: converts one type of energy to another
  - Microphone
  - Speaker
  - Thermocouple
  - Accelerometer
  - Photoresistor
  - LEDs
  - Motors….

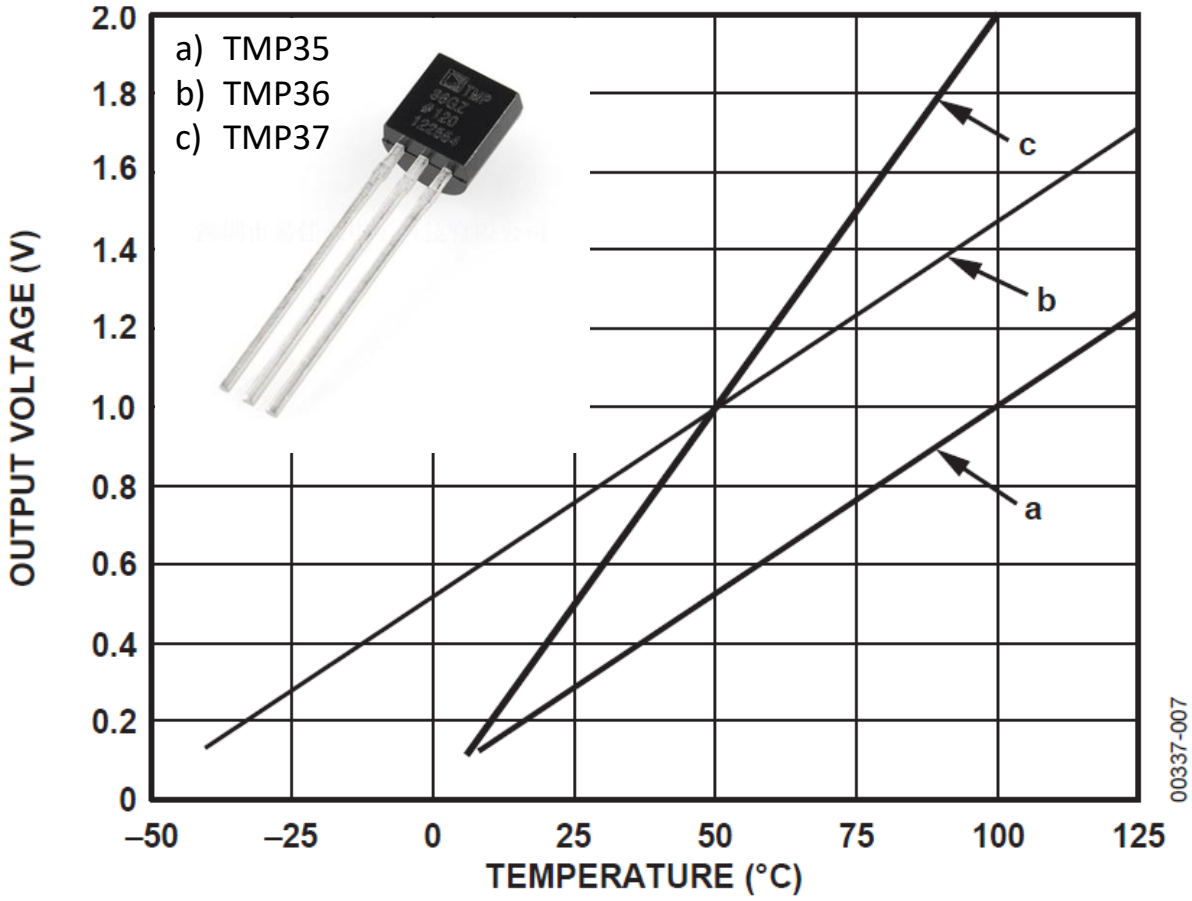




Transducers Convert One Form of Energy into Another

- Sensors convert real-world phenomena into electrical signals (like a temperature sensor)
- Actuators convert electrical signals into real-world phenomena (like an LED)



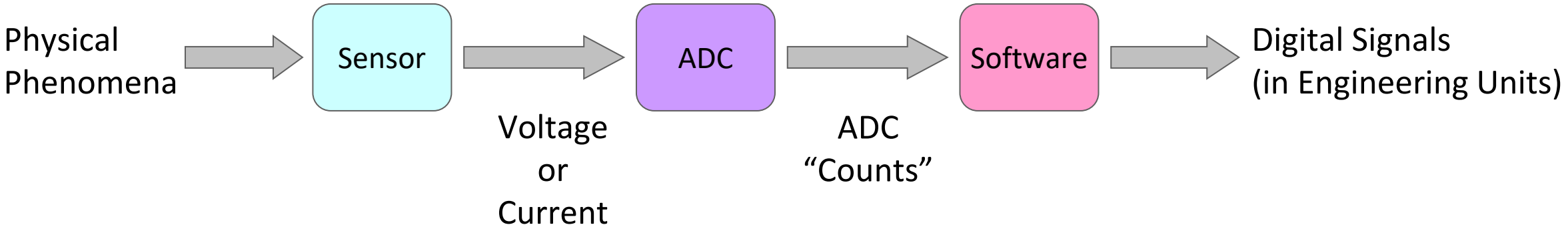


https://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf

Going from an Analog Real-World Information to Digital Signals

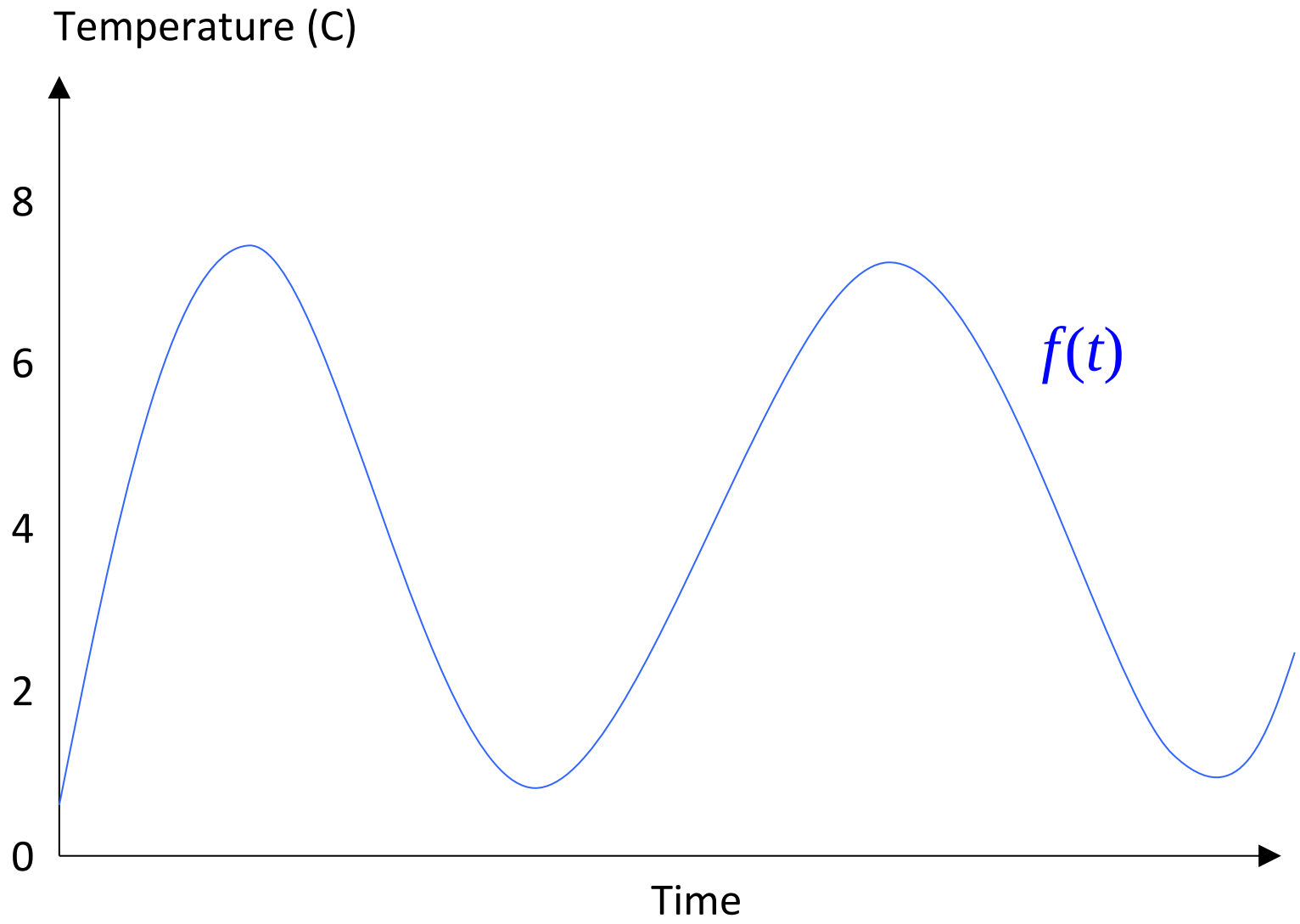
- What we want:

- How we have to get there:



Most Real-World Phenomena: Analog **AND** Continuous-Time

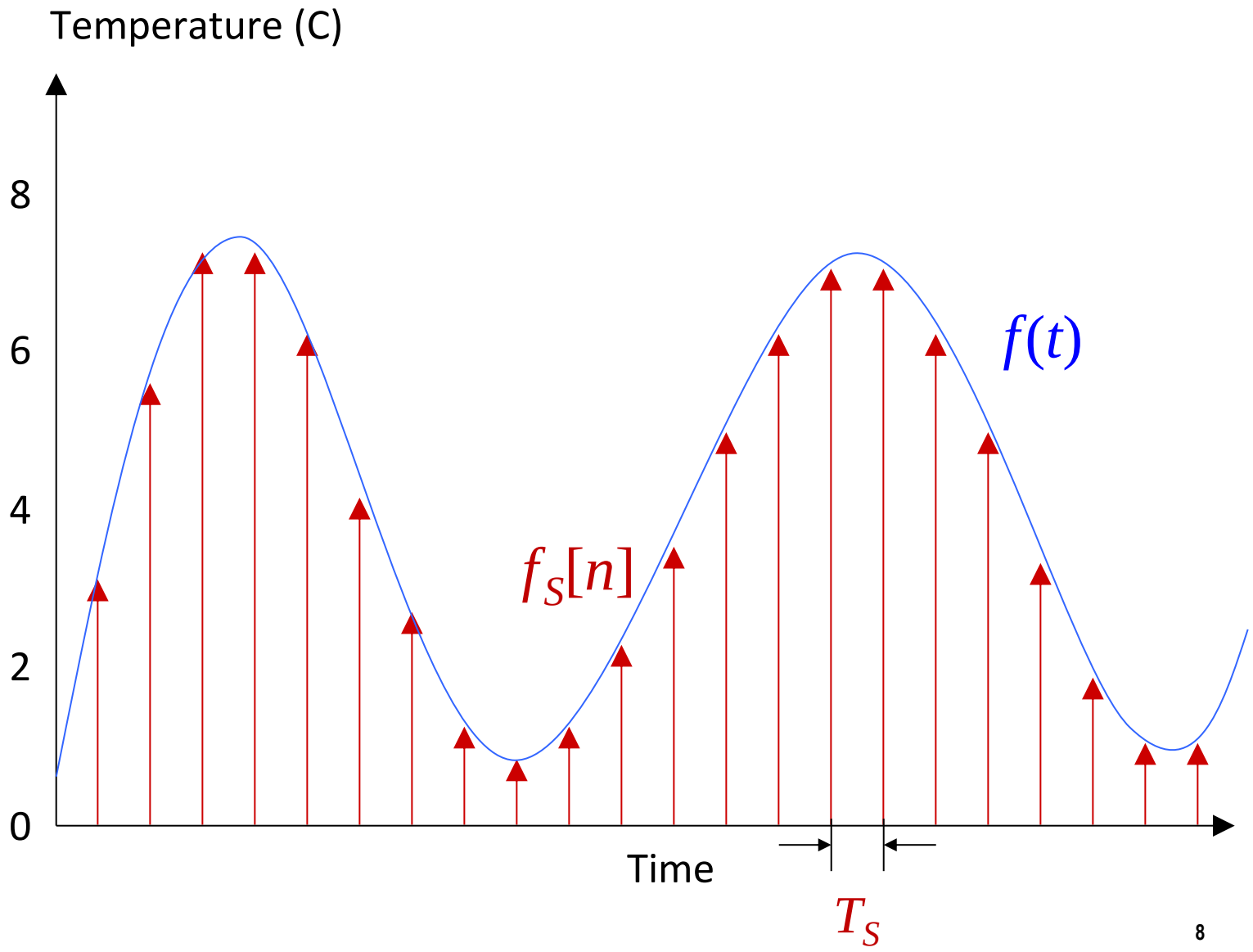
- x-axis is often time
- y-axis is often amplitude
- Both are often analog and continuous signals



Most Real-World Phenomena: Analog **AND** Continuous-Time

- x-axis is often time
- y-axis is often amplitude
- Both are often analog and continuous signals

- Sample every T_S seconds to create a discrete-time signal

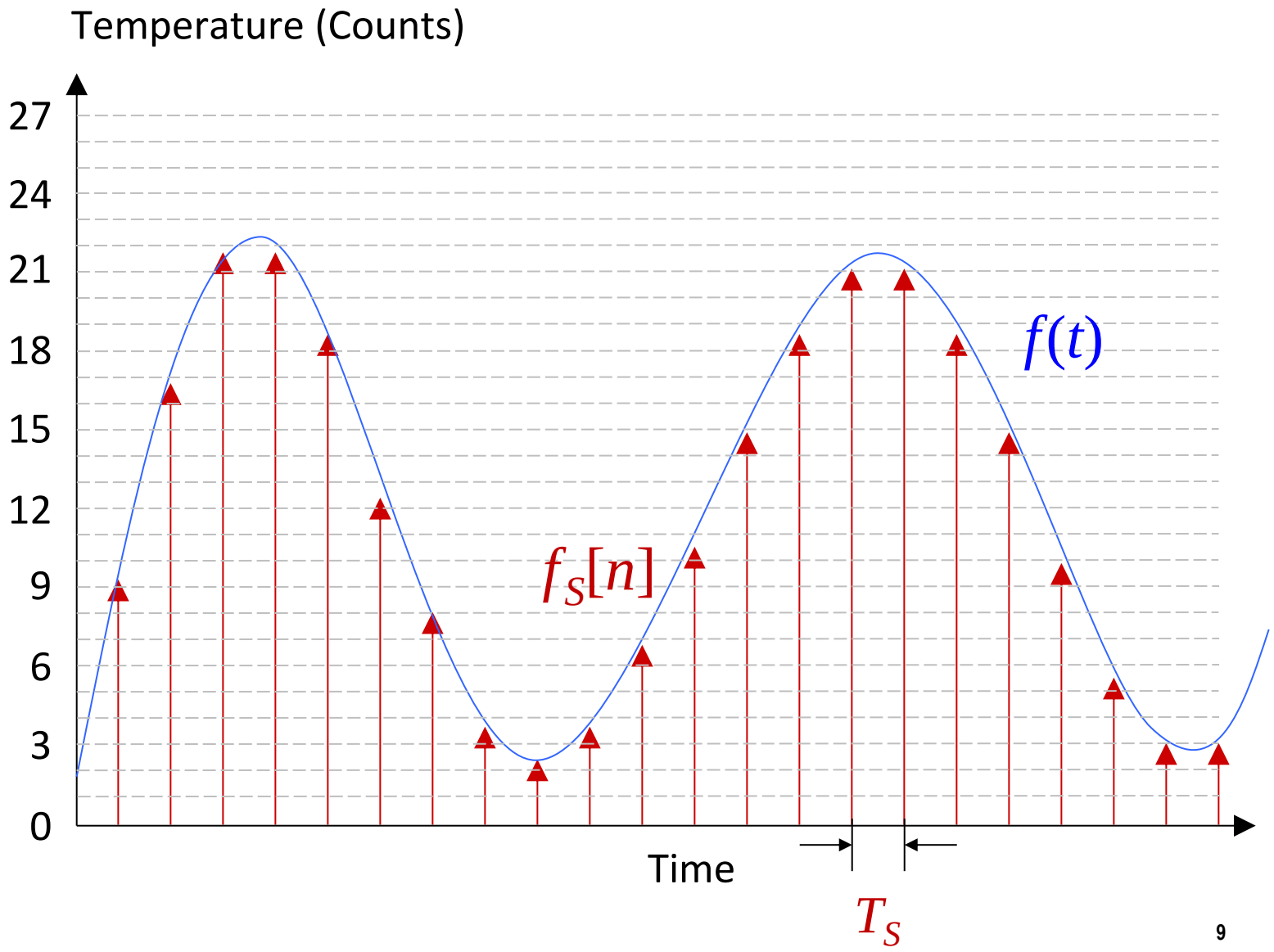


Most Real-World Phenomena: Analog **AND** Continuous-Time

- x-axis is often time
- y-axis is often amplitude
- Both are often analog and continuous signals

- Sample every T_S seconds to create a discrete-time signal

- Sample amplitude with a finite step-size to create a digital, discrete-value signal

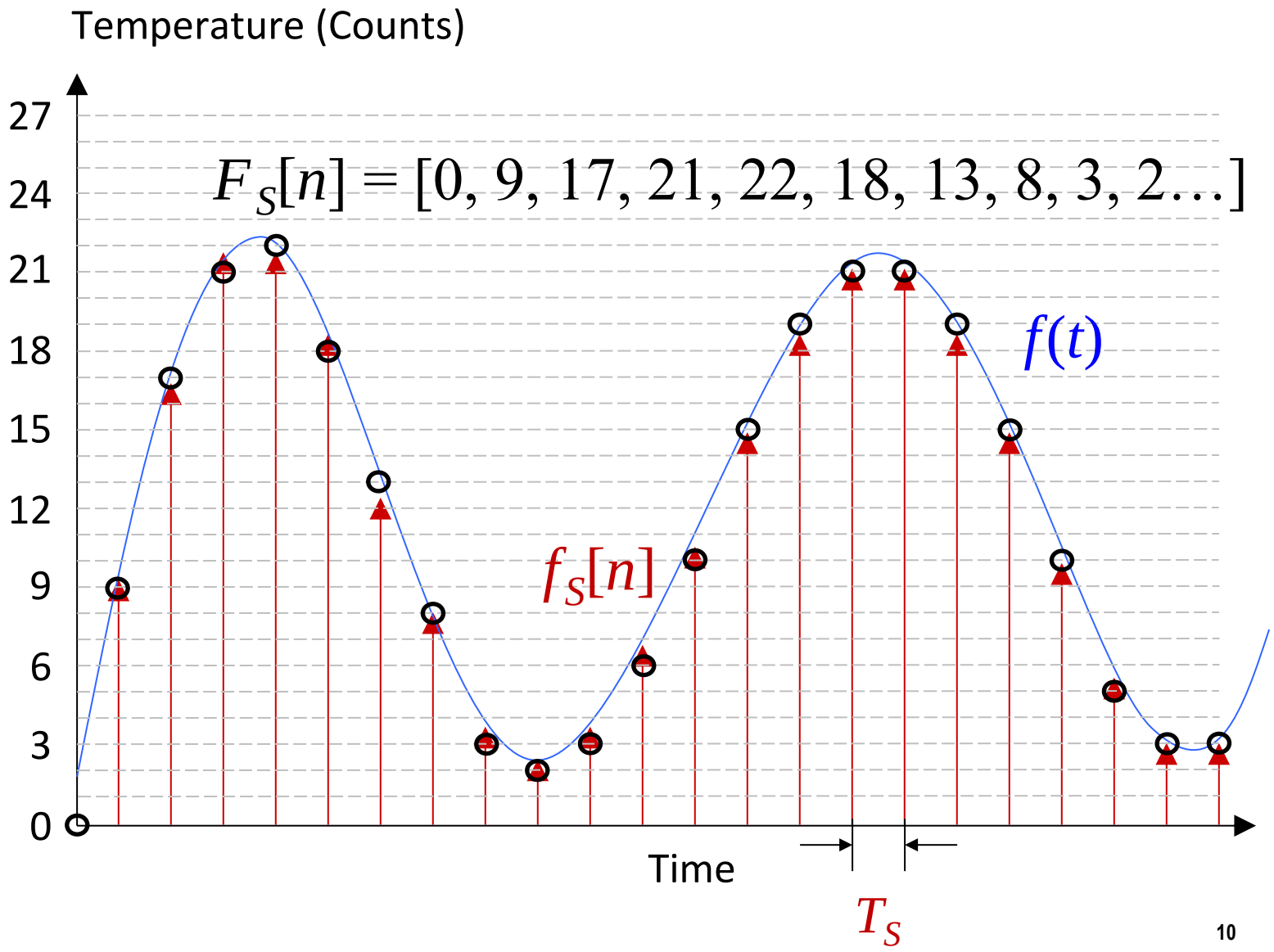


Most Real-World Phenomena: Analog **AND** Continuous-Time

- x-axis is often time
- y-axis is often amplitude
- Both are often analog and continuous signals

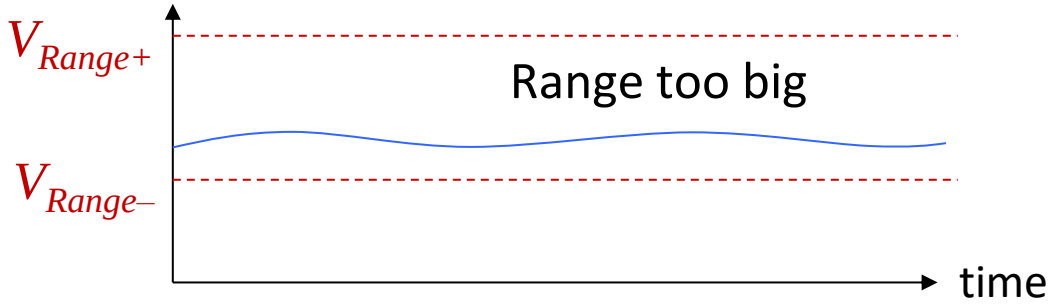
- Sample every T_S seconds to create a discrete-time signal

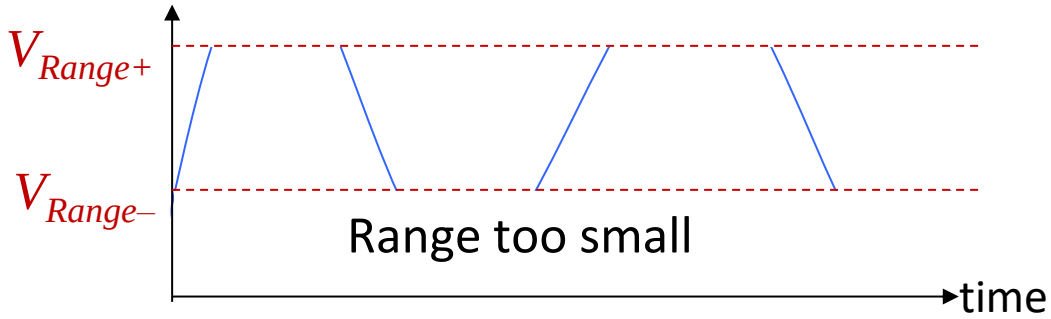
- Sample amplitude with a finite step-size to create a digital, discrete-value signal

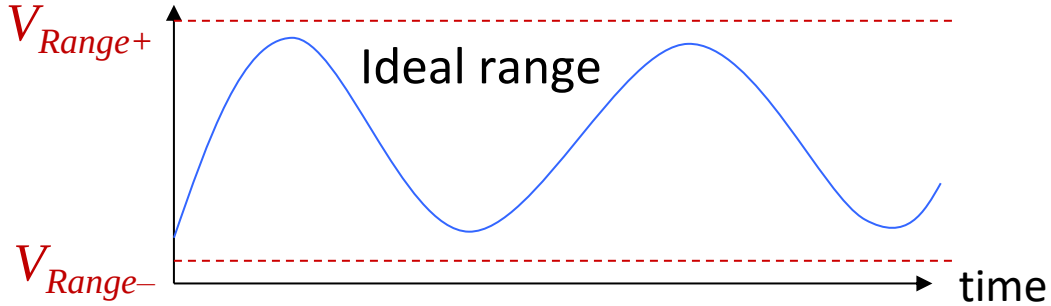


Choosing the Right Sensor Range

- The sensor's (or sensor's circuit) has a specified output voltage range
- The ADC has a specified voltage range for its input V_{Range-} (or V_{r-}) to V_{Range+} (or V_{r+})
  - V_{Range-} is usually 0V, but some ADCs allow this to be changed to other values
  - V_{range+} is usually V_{DD} (like 3.3V), but some ADCs allow this to be changed to other values
- Usually want a little margin



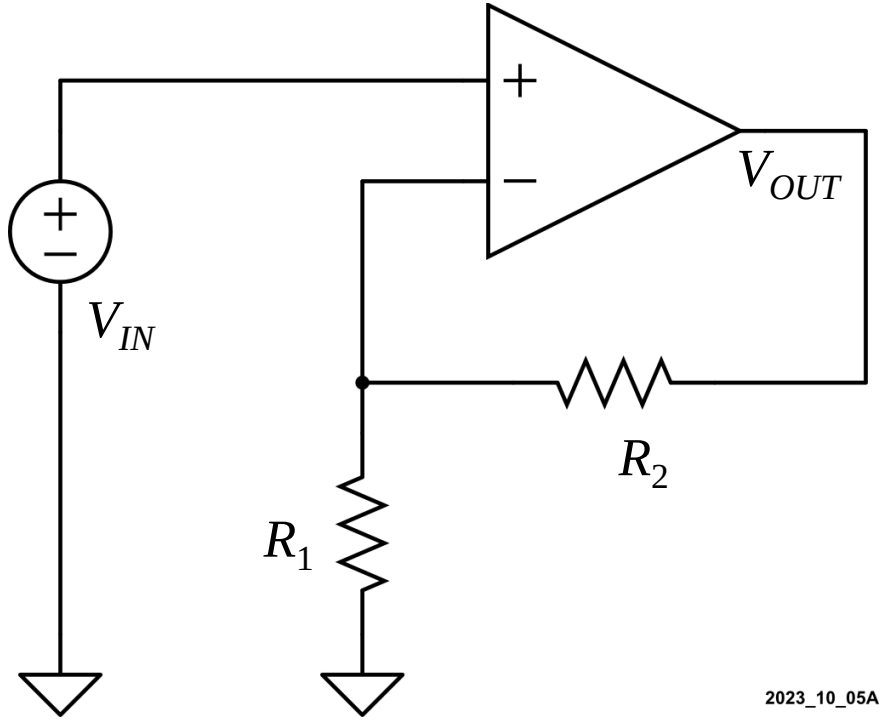




Signal Conditioning with Operational Amplifiers (Op Amps)

- Usually, op amps are modeled as ideal devices with negative feedback:
  - The non-inverting (V_+) and inverting (V_-) inputs will be the same voltage
  - The current flowing into the V_+ and V_- inputs is 0A
  - The output voltage will adjust to make the above two rules true
- Non-inverting amplifiers can make smaller voltages larger
- Other configurations have different transfer functions

$$\frac{V_{OUT}}{V_{IN}} = 1 + \frac{R_2}{R_1}$$



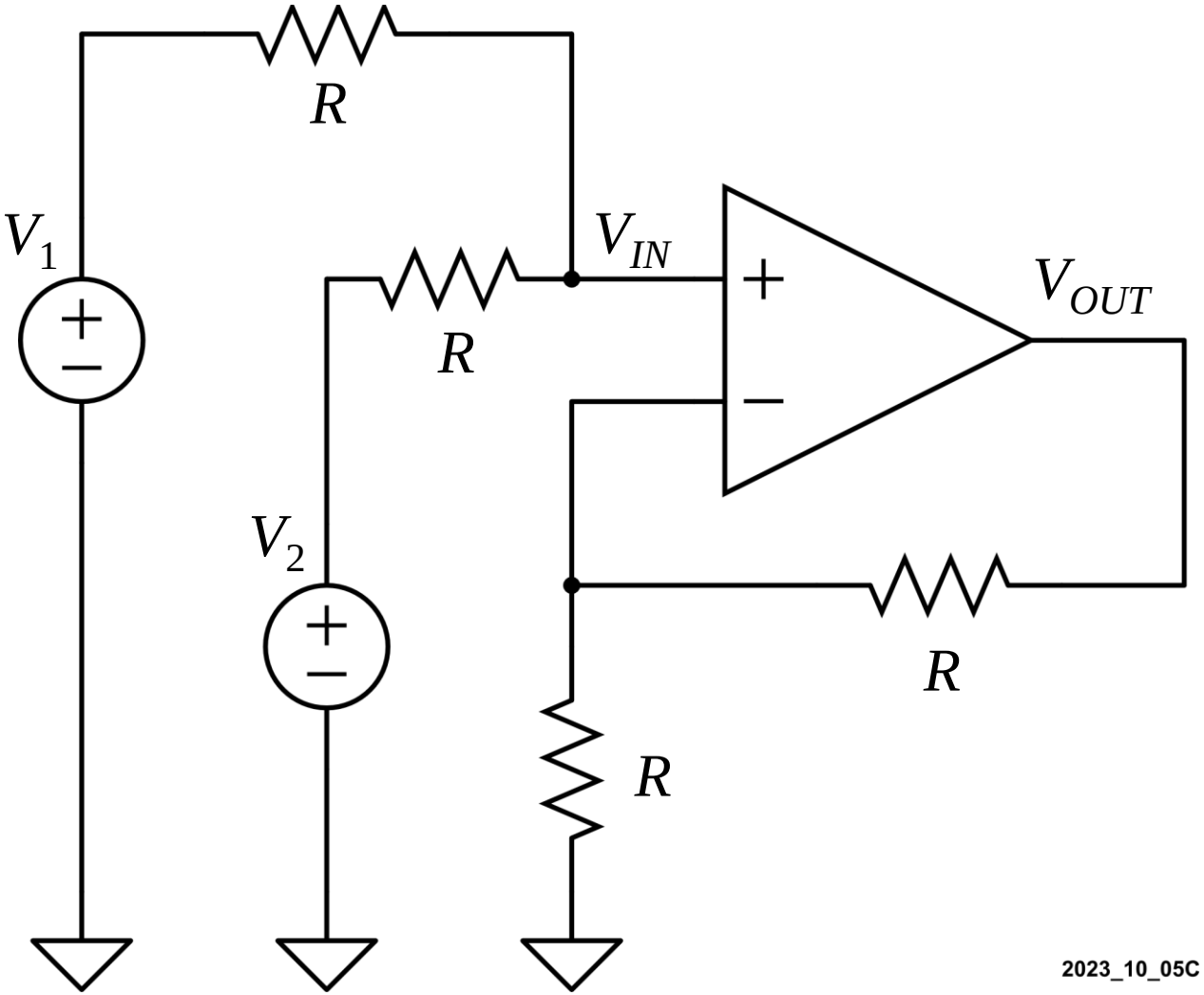
Add a DC Offset to Small Voltages with a Summing Amplifier Op Amp Circuit

- Add two (or more) voltages

$$V_{IN} = \frac{V_1 + V_2}{2}$$

$$V_{OUT} = V_{IN}\left(1 + \frac{R}{R}\right) = 2V_{IN}$$

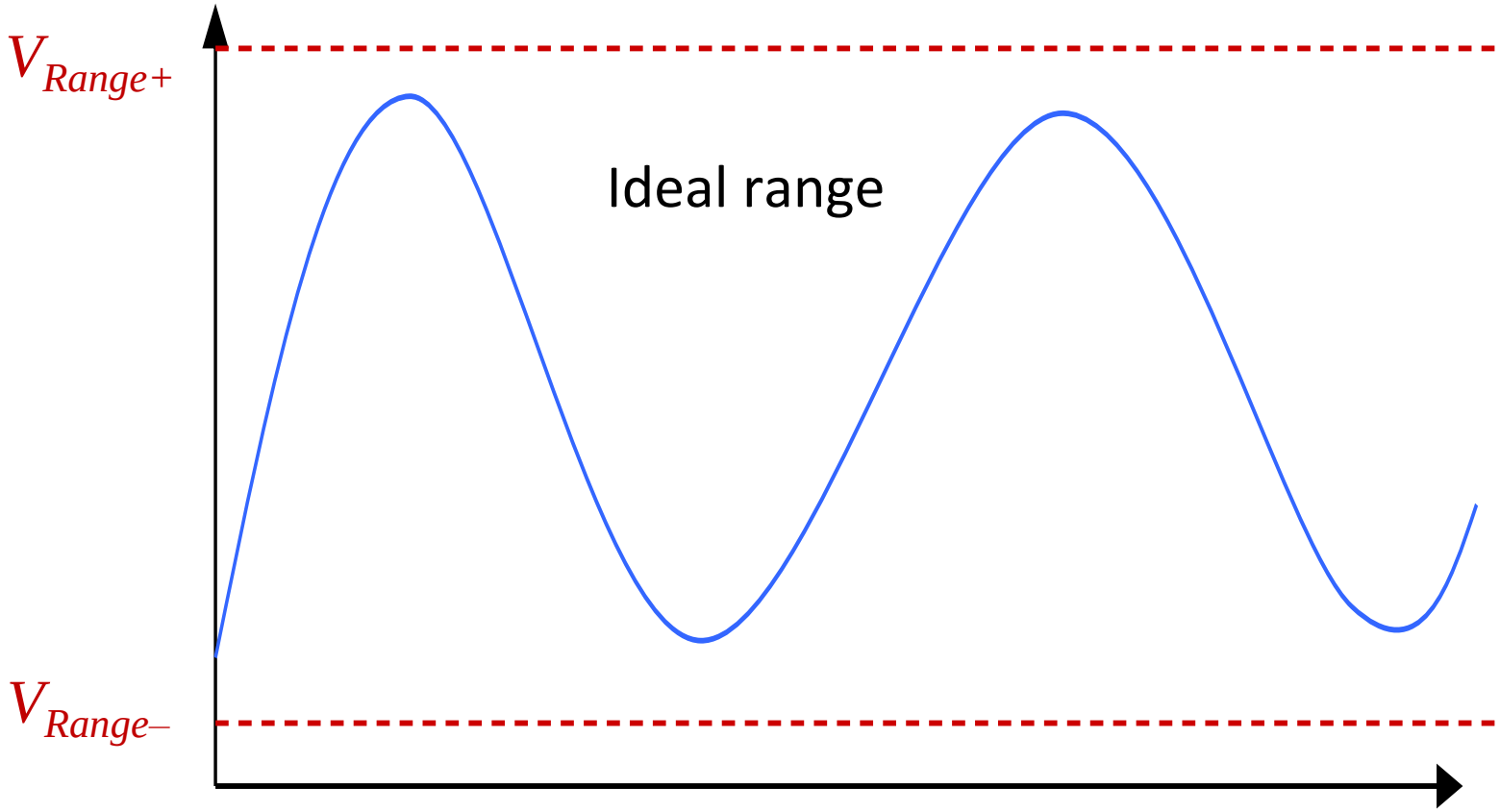
$$V_{OUT} = 2\left(\frac{V_1 + V_2}{2}\right) = V_1 + V_2$$



2023_10_05C

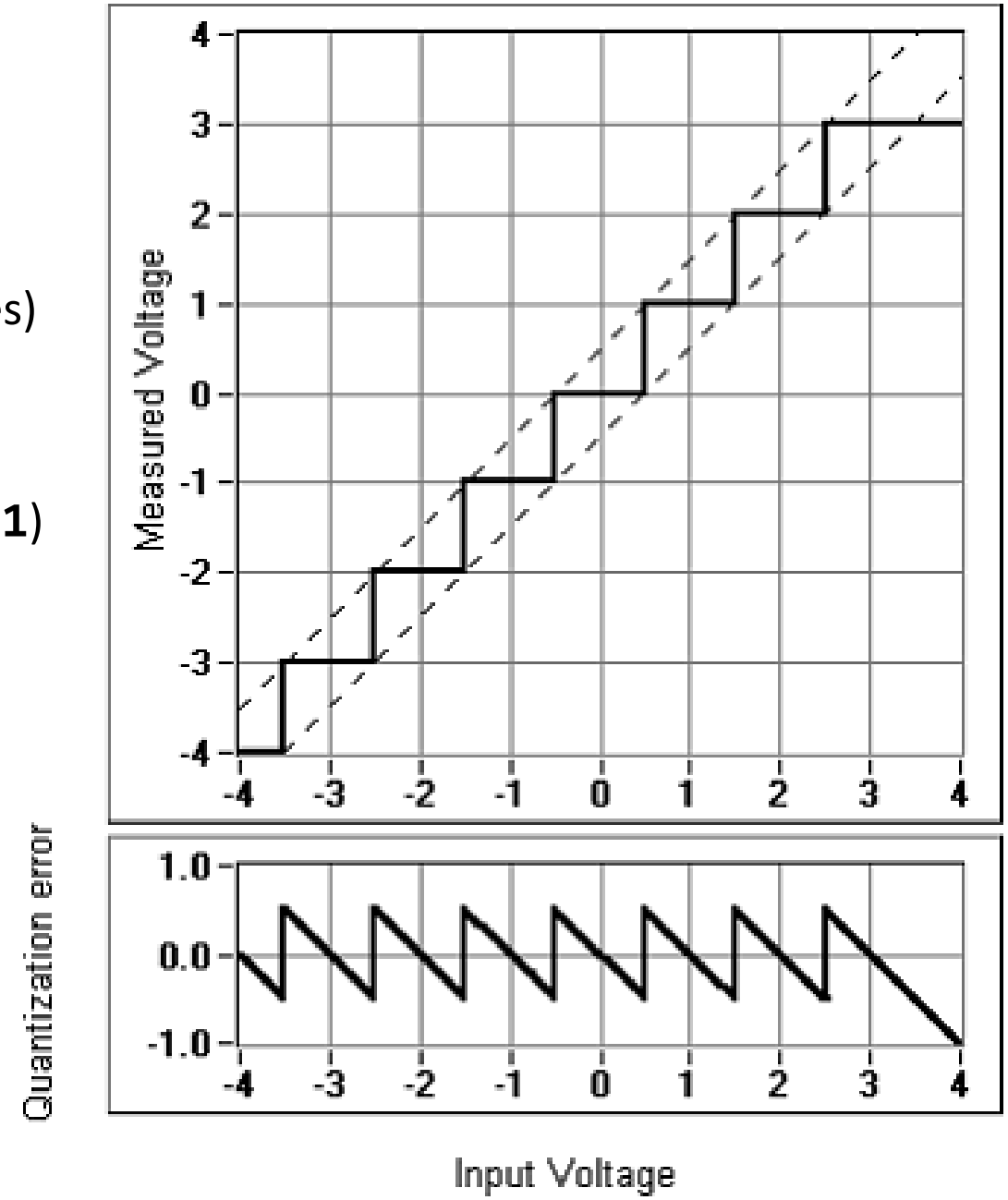
Condition the Sensor Output Voltage by Setting the ADC Range and DC Offset (with a Summing Amplifier)

- Adjust the ADC's range
- Maximize the signal's maximum and minimum values inside the ADC range
- Might require you to try a few different options to optimize



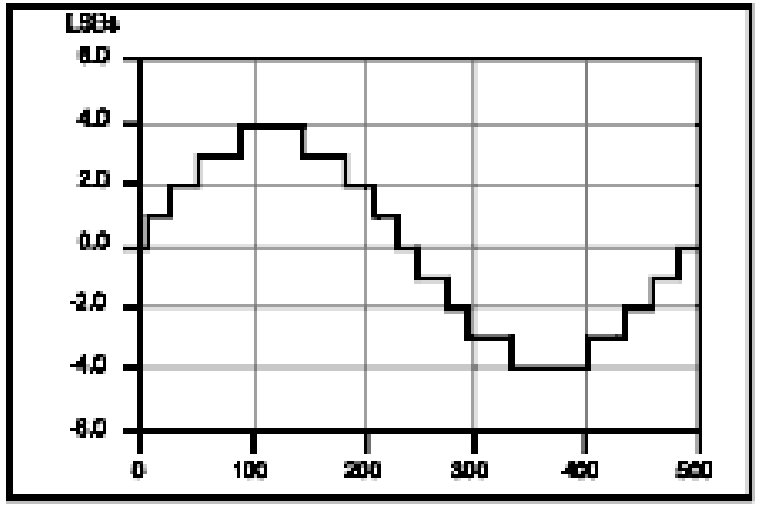
Amplitude Sampling Granularity

- Resolution: the range of analog values represented by a discrete digital value (infinite voltages → finite binary values)
- Consider a 12-bit ADC
  - 2^{12} = 4096 values (**000000000000** through **111111111111**)
  - Each binary value (step) represents $(V_{r+} - V_{r-}) / 4096$
  - For example, $(3.3\text{V} - 0\text{V}) / 4096 \text{ steps} = 806\mu\text{V} / \text{step}$
- Larger range $(V_{r+} - V_{r-})$ ➞ less information
- Larger range $(V_{r+} - V_{r-})$ ➞ more quantization error
- More bits ➞ more information

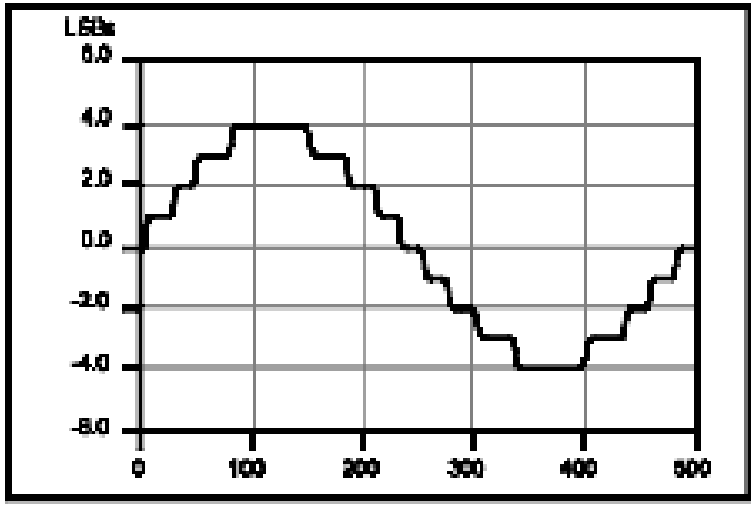


Reduce Quantization Errors with Dithering

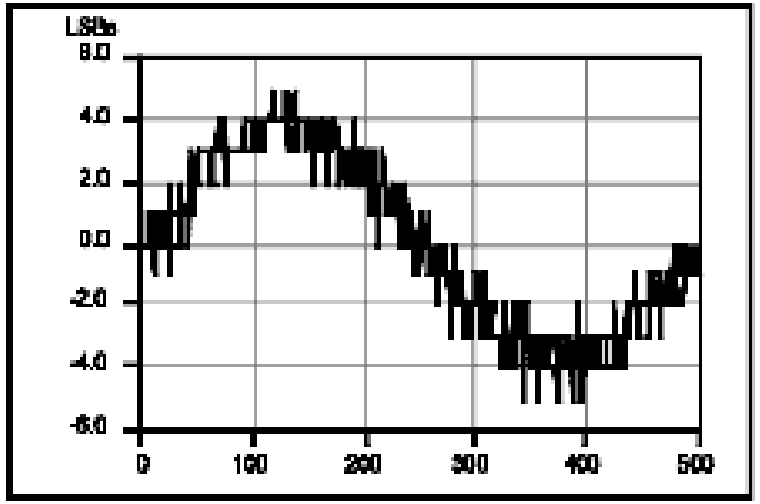
- Quantization errors can result in large-scale patterns that don't accurately describe the analog signal
- Oversample and dither (introduce white noise to randomize the quantization error
- Many general-purpose microcontroller boards have white noise generation capability for this very reason



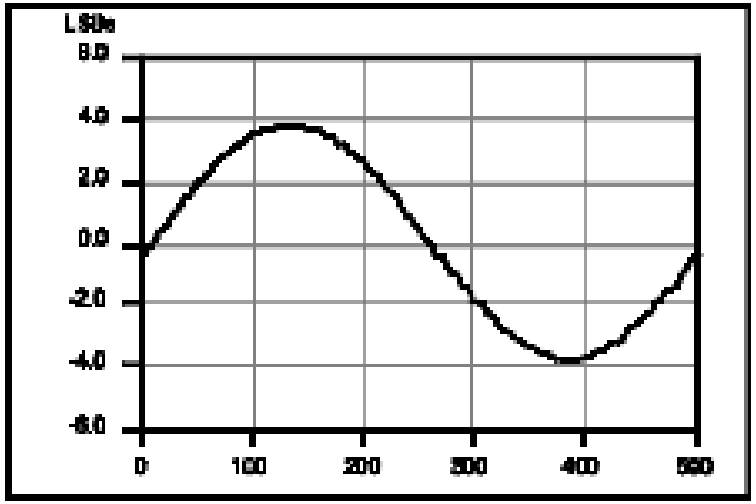
a. Dither disabled; no averaging



b. Dither disabled; average of 50 acquisitions



c. Dither enabled; no averaging

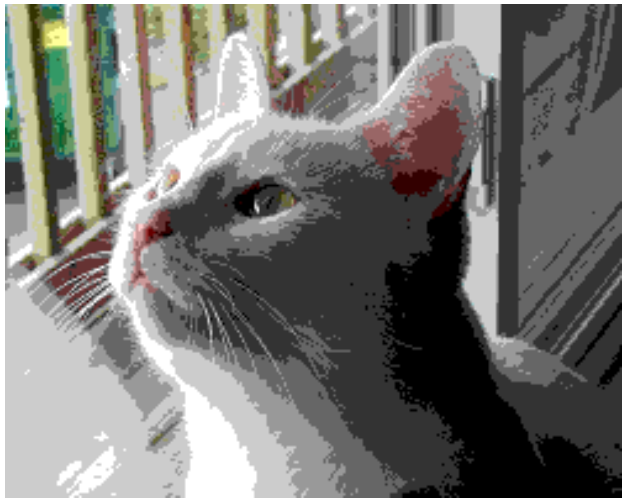


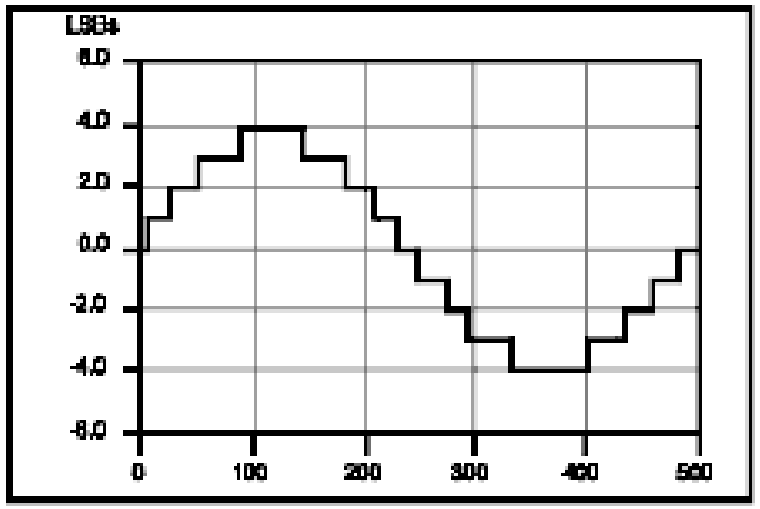
d. Dither enabled; average of 50 acquisitions

Electrical & Computer ENGINEERING

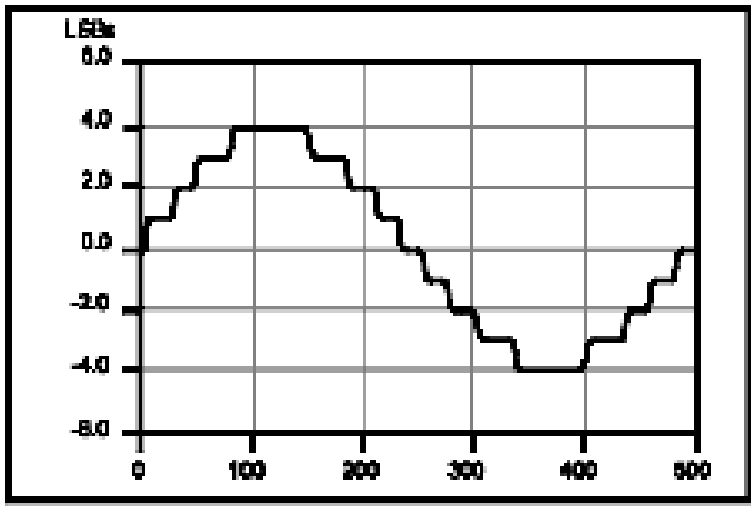
Reduce Quantization Errors with Dithering

Dithering disabled



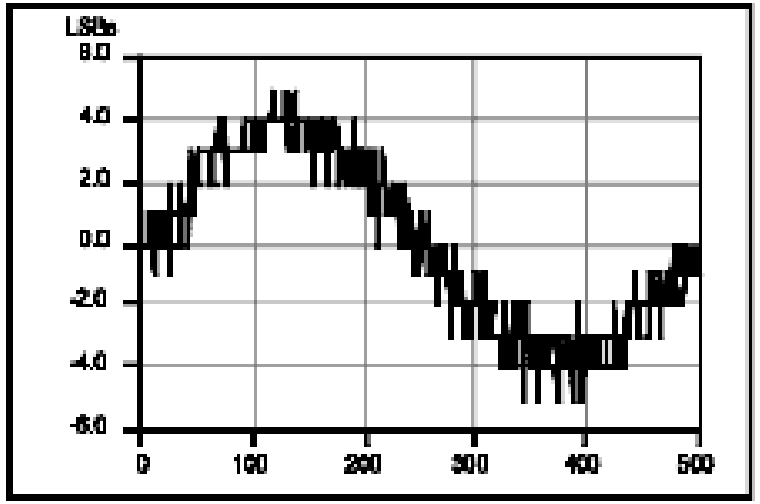


a. Dither disabled; no averaging

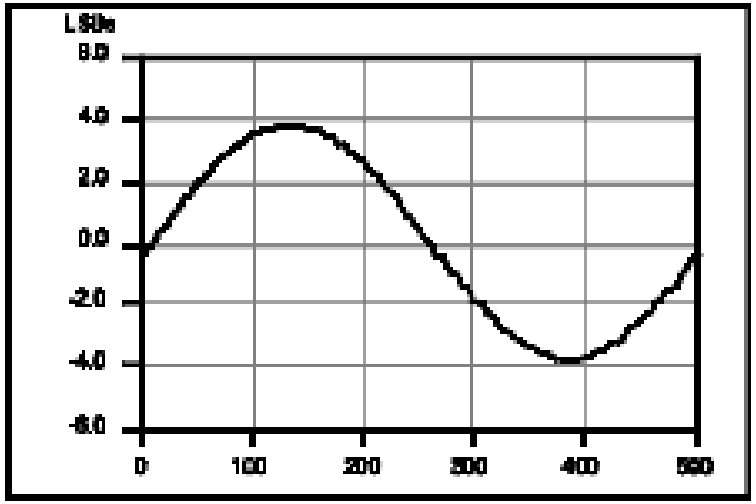


b. Dither disabled; average of 50 acquisitions

Dithering enabled



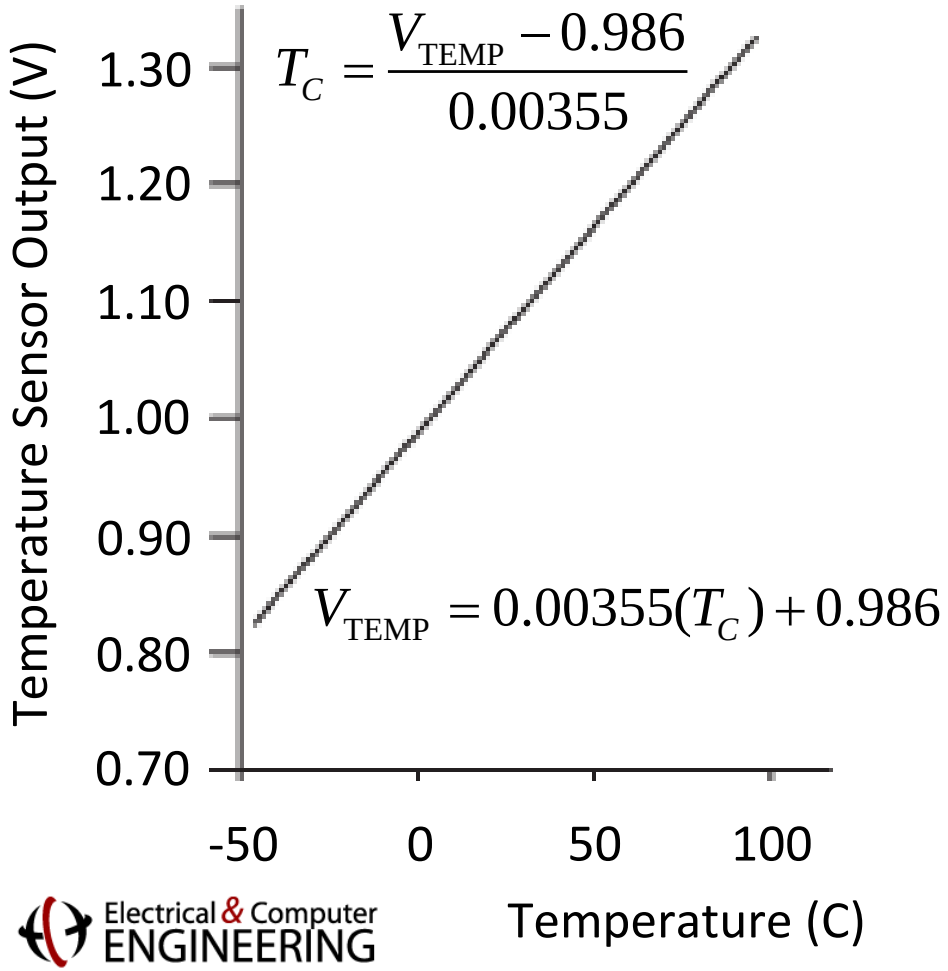
c. Dither enabled; no averaging



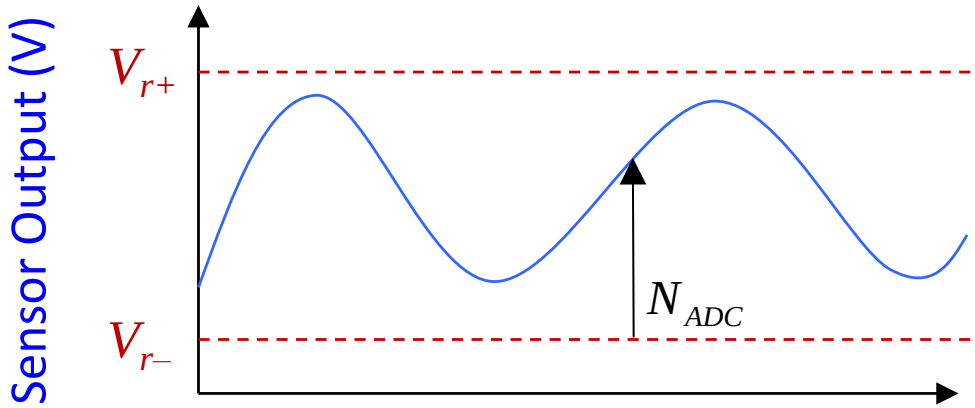
d. Dither enabled; average of 50 acquisitions

Converting Between Temperature, Voltage, ADC Counts, and Engineering Units

- Converting temperature to voltage



- Converting voltage to ADC counts or steps (N_{ADC})



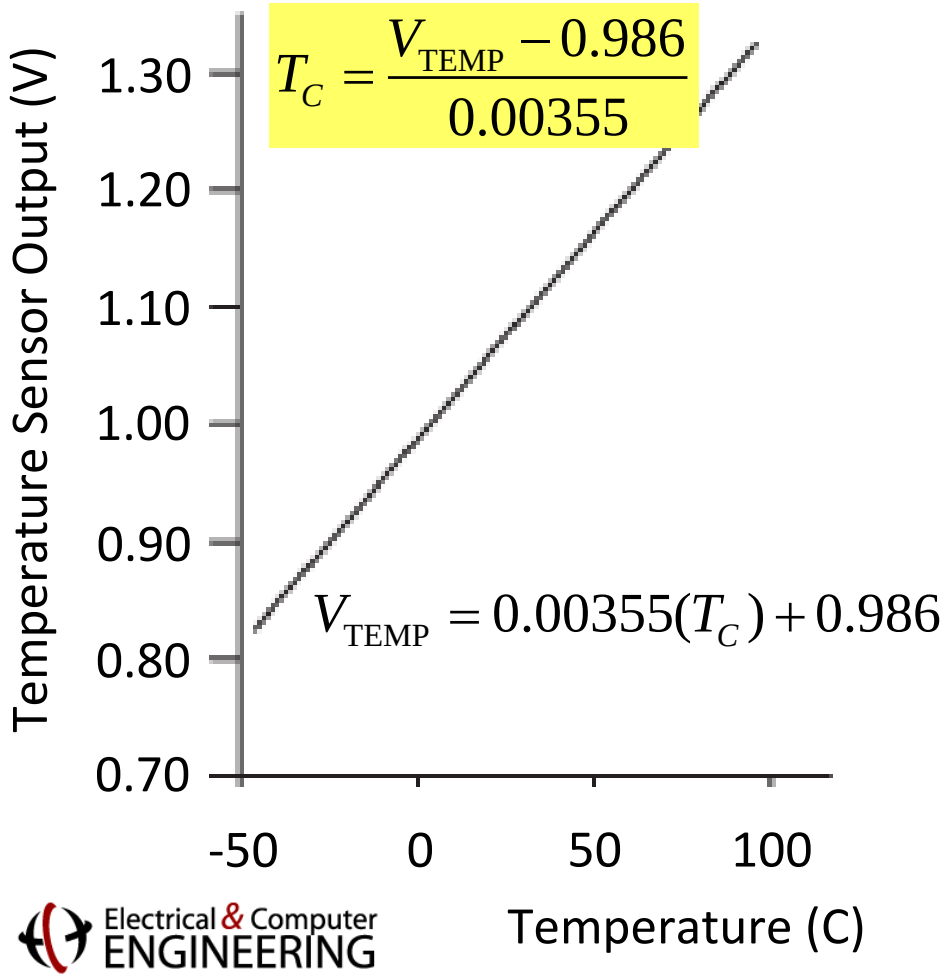
$$N_{ADC} = \left(2^n - 1\right)\left(\frac{V_{IN} - V_{r-}}{V_{r+} - V_{r-}}\right) = \left(2^{12} - 1\right)\left(\frac{V_{IN} - V_{r-}}{V_{r+} - V_{r-}}\right) = 4095\left(\frac{V_{IN} - V_{r-}}{V_{r+} - V_{r-}}\right)$$

- Converting ADC counts to engineering units
  - Convert V_{TEMP} (between 0 and 4095) back into temperature (-40C and +100C)

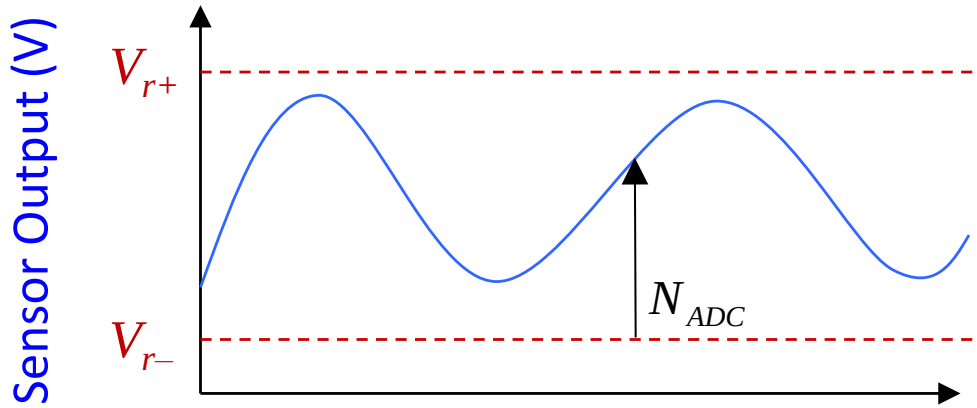
Electrical & Computer ENGINEERING

Converting Between Temperature, Voltage, ADC Counts, and Engineering Units

- Converting temperature to voltage



- Converting voltage to ADC counts or steps (N_{ADC})



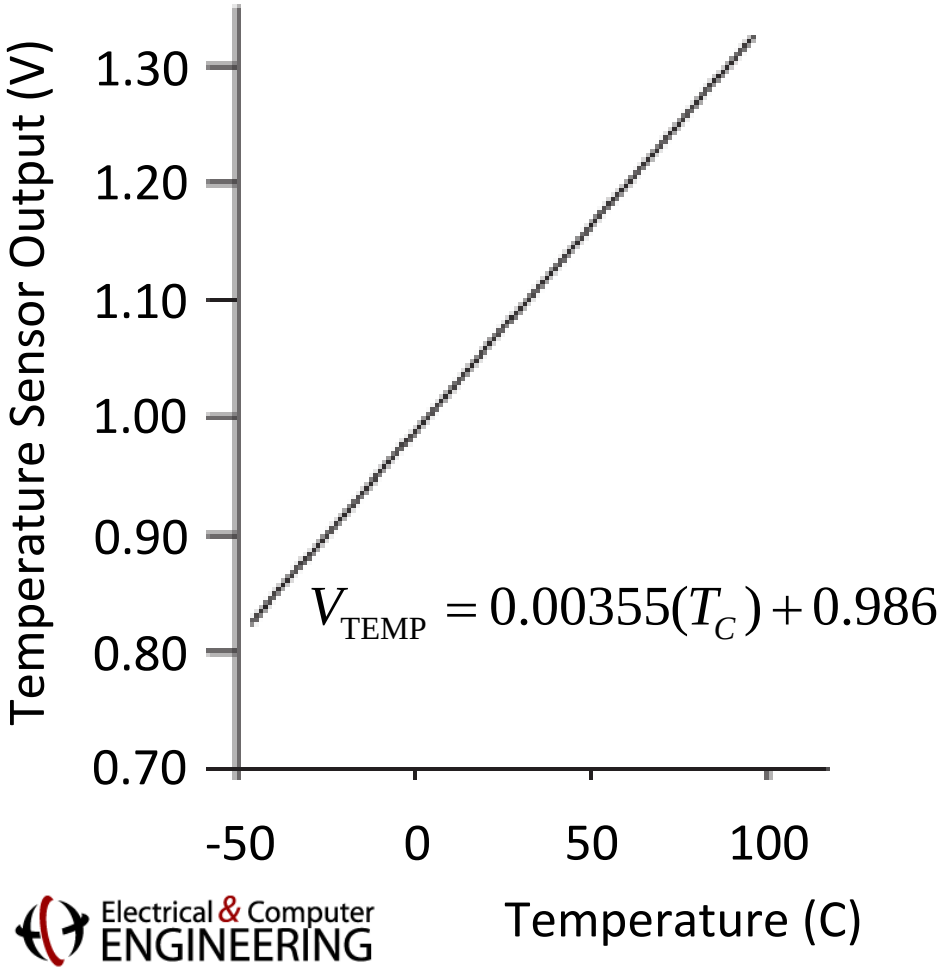
$$N_{ADC} = (2^n - 1)\left(\frac{V_{IN} - V_{r-}}{V_{r+} - V_{r-}}\right) = (2^{12} - 1)\left(\frac{V_{IN} - V_{r-}}{V_{r+} - V_{r-}}\right) = 4095\left(\frac{V_{IN} - V_{r-}}{V_{r+} - V_{r-}}\right)$$

- Converting ADC counts to engineering units
  - Convert N_{ADC} (between 0 and 4095) back into temperature (-40C and +100C)

Electrical & Computer ENGINEERING

Example: Converting ADC Counts into Engineering Units

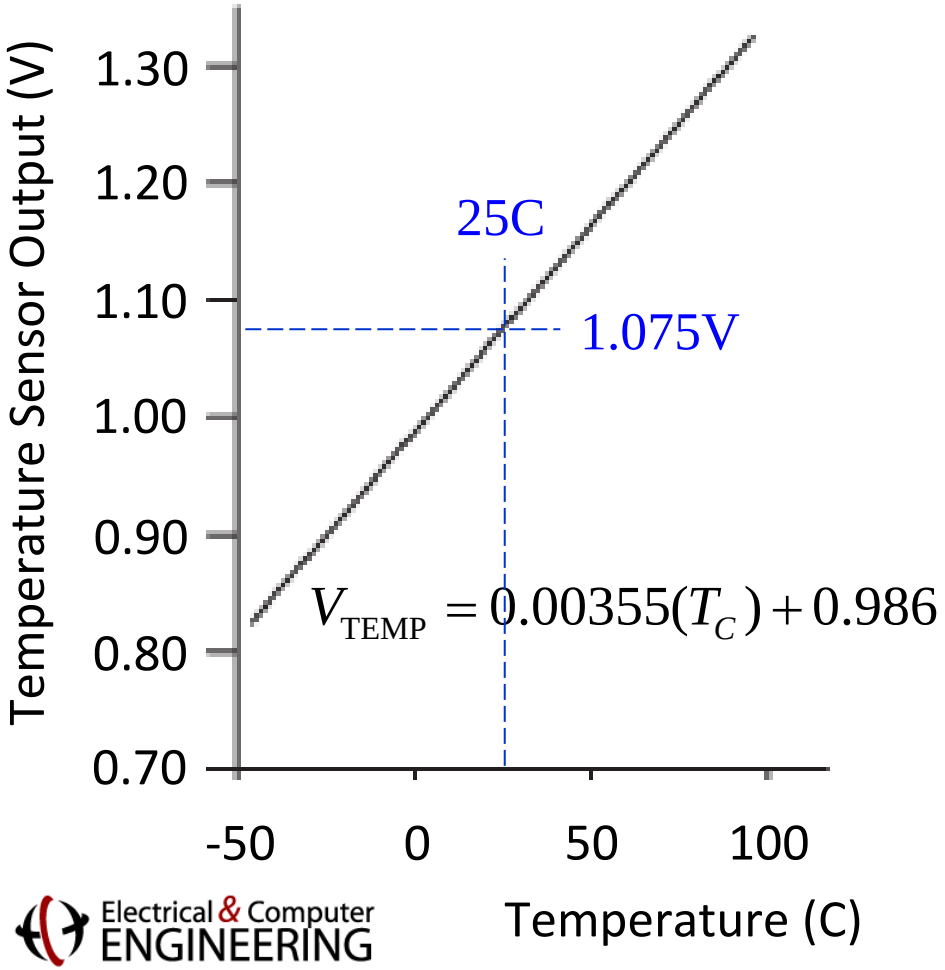
- Sensor: $0.8\text{V} < V_{OUT} < 1.35\text{V}$
- Choose $V_{r+} = 1.500\text{V}$ and $V_{r-} = 0.000\text{V}$



Electrical & Computer ENGINEERING

Example: Converting ADC Counts into Engineering Units

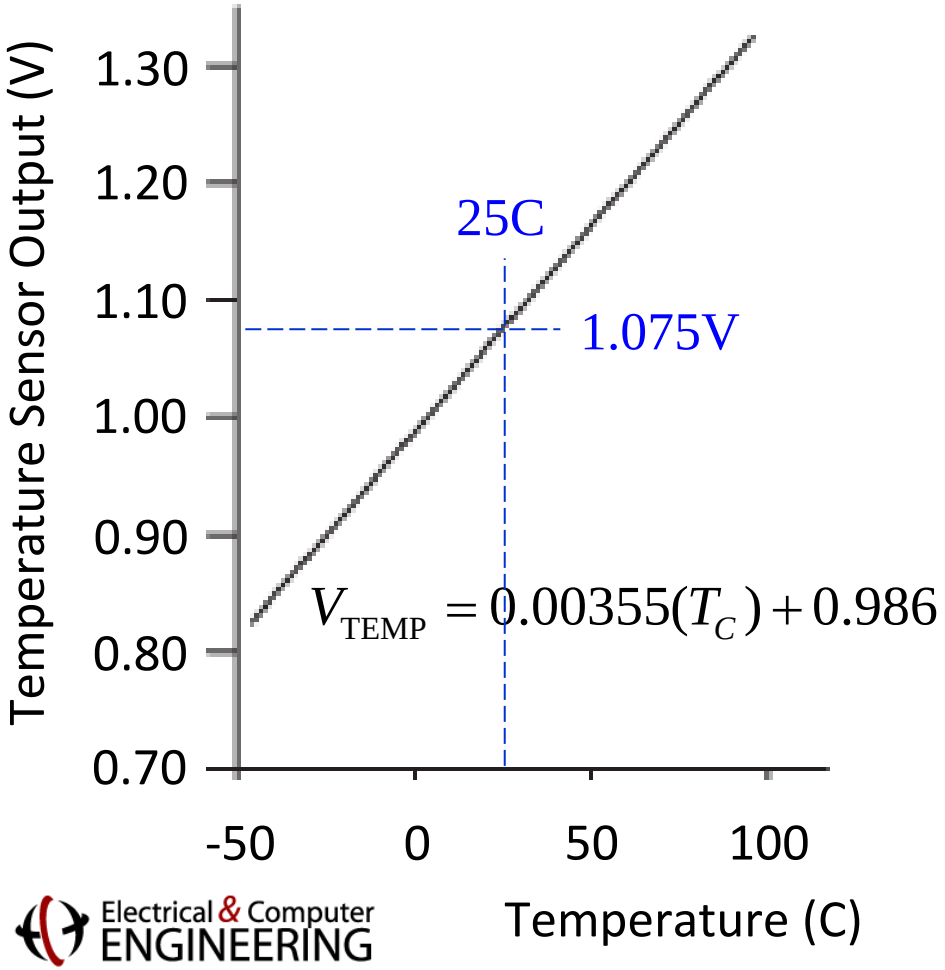
- Sensor: $0.8\text{V} < V_{OUT} < 1.35\text{V}$
- Choose $V_{r+} = 1.500\text{V}$ and $V_{r-} = 0.000\text{V}$



Electrical & Computer ENGINEERING

Example: Converting ADC Counts into Engineering Units

- Sensor: $0.8\text{V} < V_{OUT} < 1.35\text{V}$
- Choose $V_{r+} = 1.500\text{V}$ and $V_{r-} = 0.000\text{V}$



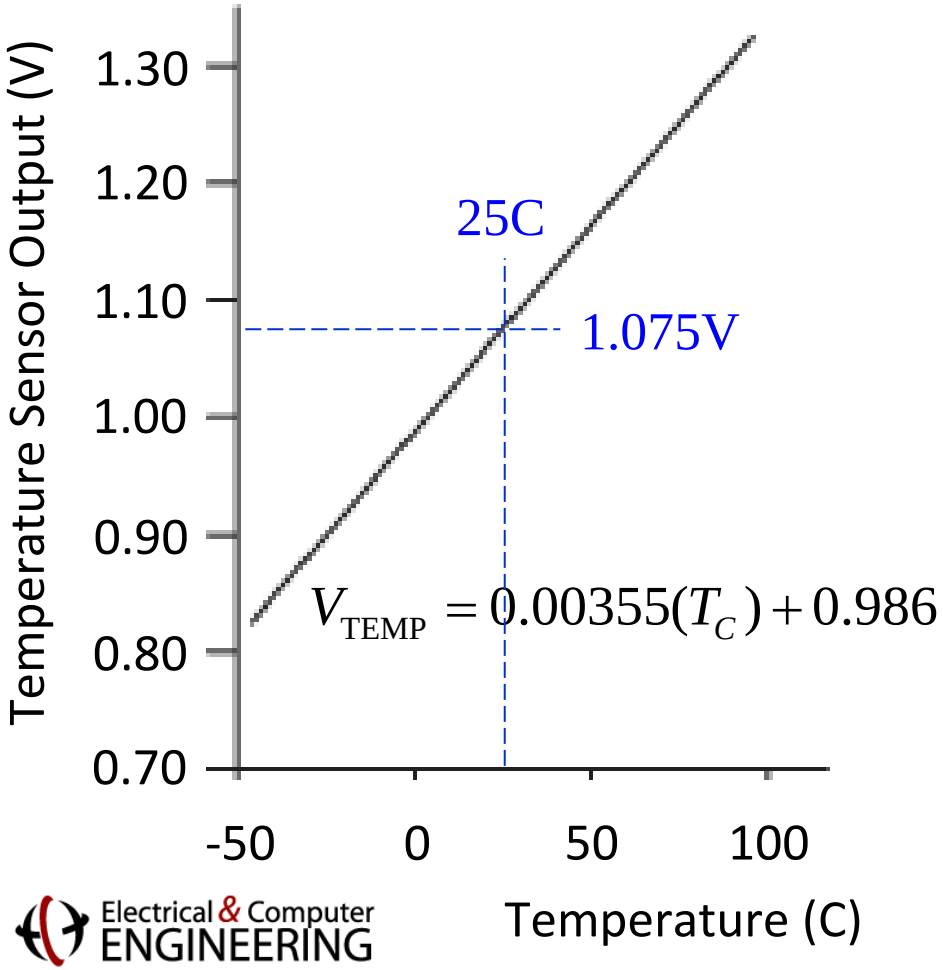
- Converting voltage to ADC counts (N_{ADC})

$$N_{ADC} = 4095\left(\frac{V_{IN} - V_{r-}}{V_{r+} - V_{r-}}\right) = 4095\left(\frac{1.075V - 0V}{1.500V - 0V}\right) = 2,934\ steps$$

Electrical & Computer ENGINEERING

Example: Converting ADC Counts into Engineering Units

- Sensor: $0.8\text{V} < V_{OUT} < 1.35\text{V}$
- Choose $V_{r+} = 1.500\text{V}$ and $V_{r-} = 0.000\text{V}$



- Converting voltage to ADC counts (N_{ADC})

$$N_{ADC} = 4095\left(\frac{V_{IN} - V_{r-}}{V_{r+} - V_{r-}}\right) = 4095\left(\frac{1.075V - 0V}{1.500V - 0V}\right) = 2,934\ steps$$

- Converting ADC counts (N_{ADC}) to temperature

$$V_{IN} = \left(\frac{N_{ADC}}{4095}\right)\left(V_{r+} - V_{r-}\right) + V_{r-}$$

$$V_{IN} = \left(\frac{2934}{4095}\right)\left(1.5V\right) + 0V = 1.075V$$

- Converting voltage back into temperature

$$T_C = \frac{V_{\text{TEMP}} - 0.986}{0.00355} = \frac{1.075 - 0.986}{0.00355} = 25C$$

Electrical & Computer ENGINEERING

Example: Converting ADC Counts into Engineering Units

- Sensor: $0.8\text{V} < V_{OUT} < 1.35\text{V}$
- Choose $V_{r+} = 1.500\text{V}$ and $V_{r-} = 0.000\text{V}$
- $T_C = 25\text{C} \rightarrow V_{OUT} = 1.075\text{V}$
- $V_{OUT} = 1.075\text{V} \rightarrow N_{ADC} = 2{,}934$ steps

- Program converts $2{,}934$ steps back to 25C

$$V_{IN} = \left(\frac{2934}{4095}\right)\left(1.5V\right) + 0V = 1.075V$$

$$T_C = \frac{V_{\text{TEMP}} - 0.986}{0.00355} = \frac{1.075 - 0.986}{0.00355} = 25C$$

```
int main() {

    int   adccount = 2934;
    float range    = 1.5;
    float vtemp;
    float tempc;

    vtemp = adccount/4095 * range;
    tempc = (vtemp-0.986)/0.00355;
    printf("vtemp: %f\n", vtemp);
    printf("tempc: %f\n", tempc);
}
```

Example: Converting ADC Counts into Engineering Units

- Sensor: $0.8\text{V} < V_{OUT} < 1.35\text{V}$
- Choose $V_{r+} = 1.500\text{V}$ and $V_{r-} = 0.000\text{V}$
- $T_C = 25\text{C} \rightarrow V_{OUT} = 1.075\text{V}$
- $V_{OUT} = 1.075\text{V} \rightarrow N_{ADC} = 2{,}934$ steps

- Program converts $2{,}934$ steps back to 25C

$$V_{IN} = \left(\frac{2934}{4095}\right)\left(1.5V\right) + 0V = 1.075V$$

$$T_C = \frac{V_{\text{TEMP}} - 0.986}{0.00355} = \frac{1.075 - 0.986}{0.00355} = 25C$$

```
int main() {

    int   adccount = 2934;
    float range    = 1.5;
    float vtemp;
    float tempc;

    vtemp = adccount/4095 * range;
    tempc = (vtemp-0.986)/0.00355;
    printf("vtemp: %f\n", vtemp);
    printf("tempc: %f\n", tempc);
}

$ gcc arithmetic.c

$ ./a.out
vtemp: 0.000000
tempc: -277.746490
```

Final Thoughts on Amplitude

- Sensor provides a range of voltages for inputs of interest to us

- Use op amps to convert those voltages into a desired ADC range of voltages
- Use the full range of the ADC, perhaps with a small margin

- Choose an ADC with enough bits to provide the resolution required ($Cost\ \alpha\ e^{\#\ of\ bits}$)
- Understand ADC and DAC errors will be a fact-of-life

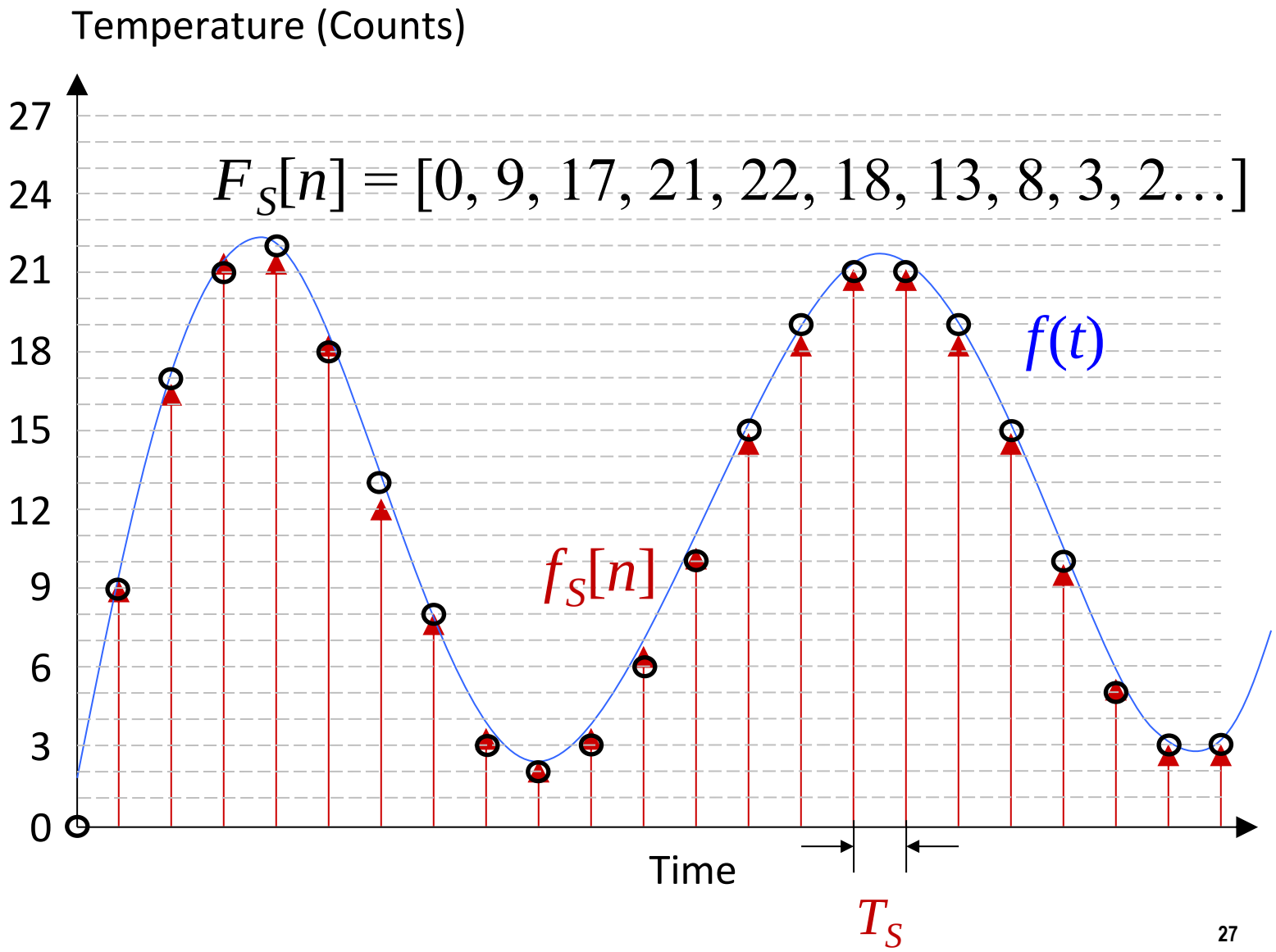
- Carefully write your code to manipulate values without losing resolution of your data
- Be careful about each arithmetic operation
- Be careful about software emulation of floating-point numbers

Continuous-Time Data and Discrete-Time Sampling

- x-axis is often time
- y-axis is often amplitude
- Both are often analog and continuous signals

- Sample every T_S seconds to create a discrete-time signal

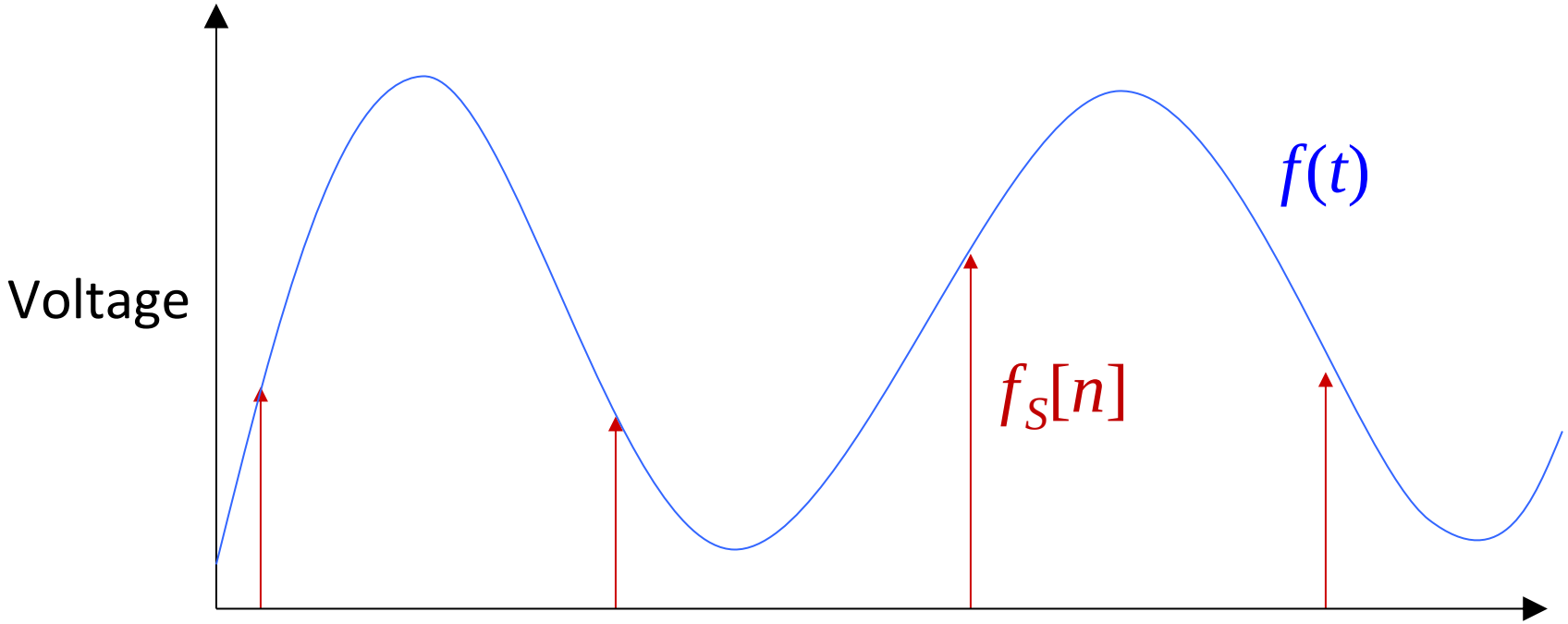
- Sample amplitude with a finite step-size to create a digital, discrete-value signal



Choosing a Sampling Rate

- ADC peripherals have a maximum sampling rate
- In addition, your program may be limited in how often it can read an ADC sample

- Sample too slowly: We can't reconstruct the signal we care about
- Sample too quickly: Waste computation, energy, resources



Nyquist-Shannon Sampling Theorem

- If a continuous-time signal contains no frequencies higher than f_{max}, it can be completely determined by discrete samples taken at a rate of

$$f_{sampling} > 2f_{max}$$

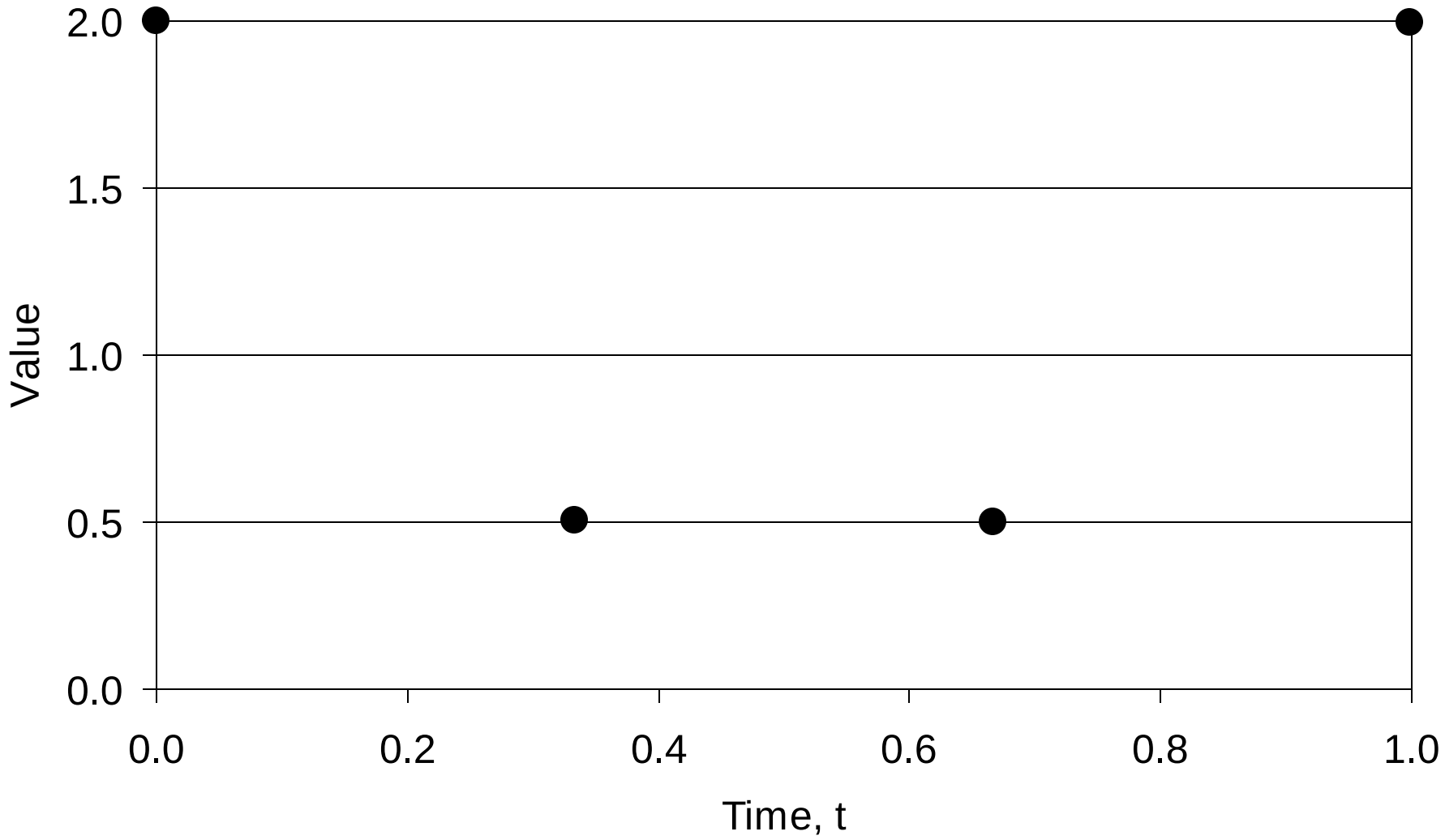
- For example, most people can "hear" audio signals between 20Hz and 20kHz
- Audio CDs: Sampled at 44.1kHz

Song sampled at 44.1kHz

Same song sampled at 1,024Hz (distorts frequencies above 512Hz)

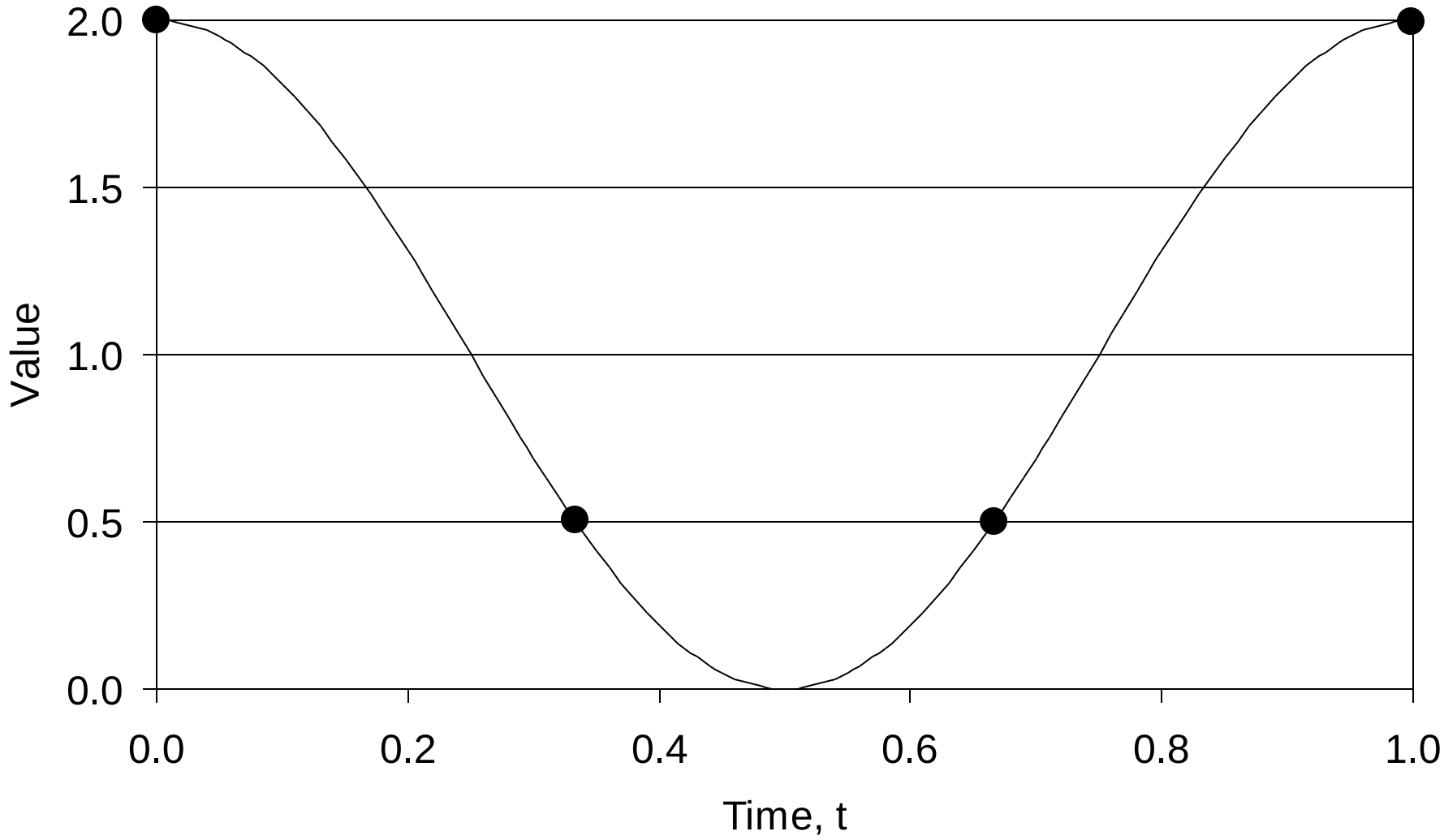
Aliasing

Samples of Different Signals Have the Same Sampled Values



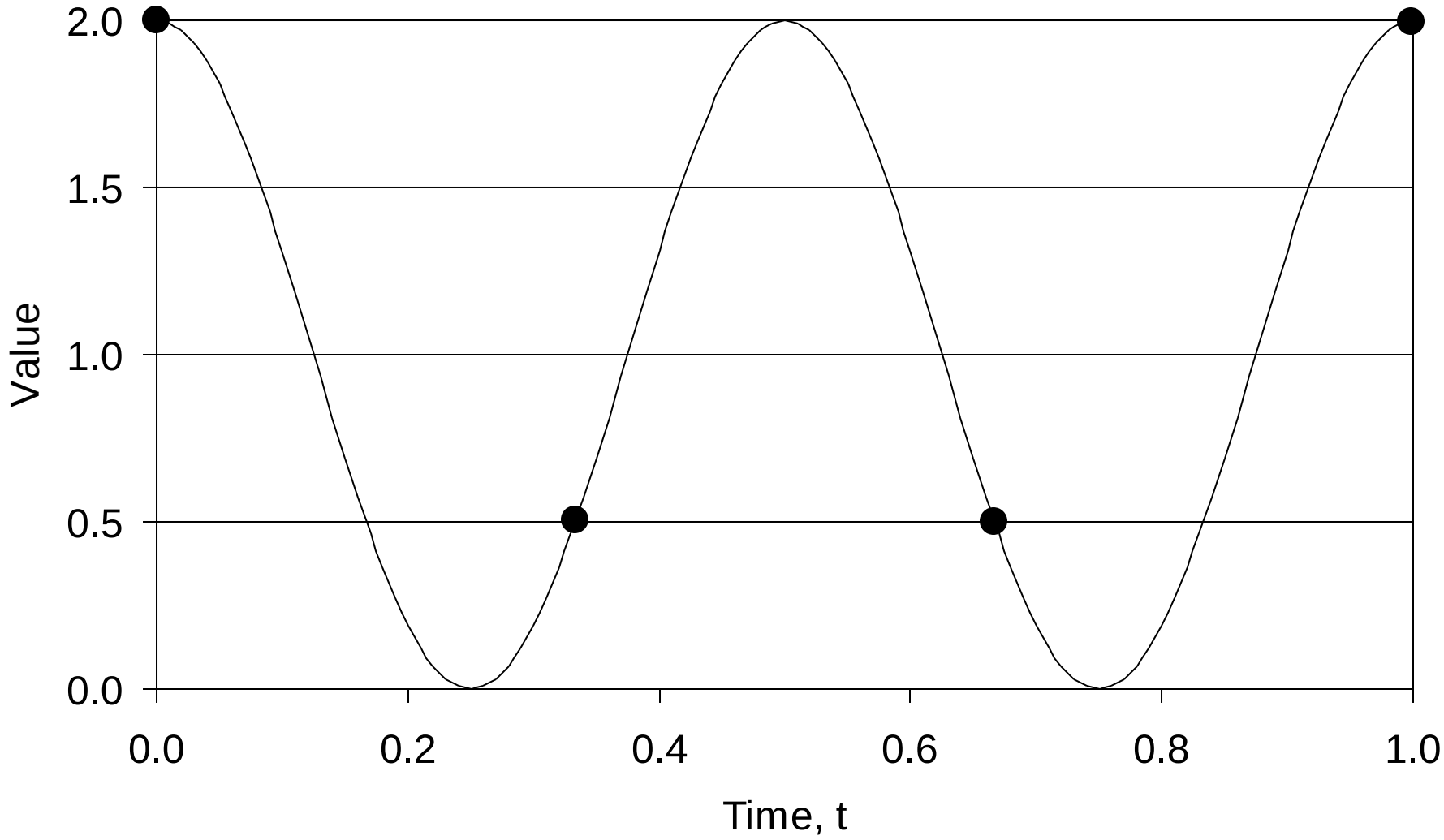
Aliasing

Samples of Different Signals Have the Same Sampled Values



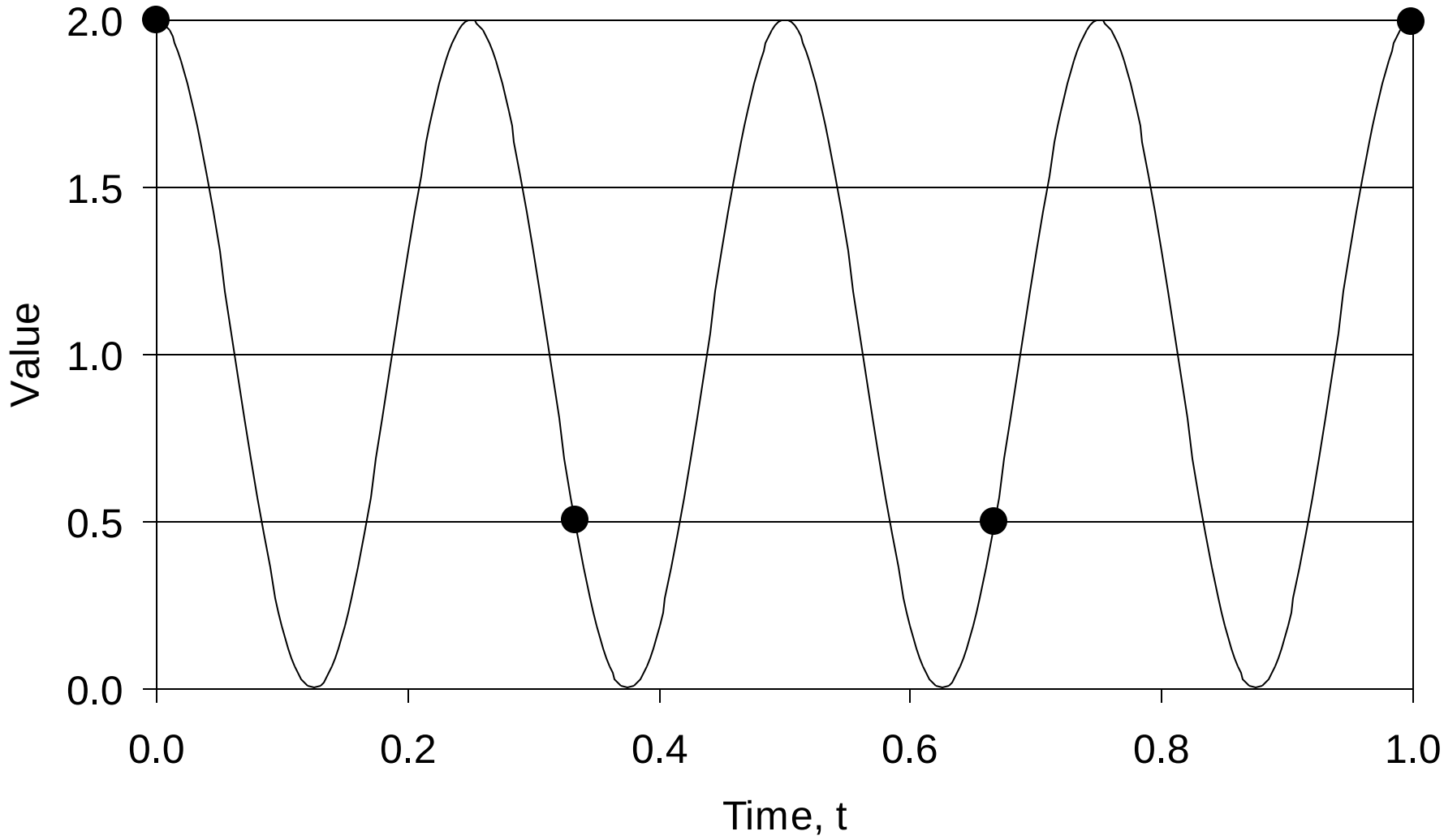
Aliasing

Samples of Different Signals Have the Same Sampled Values



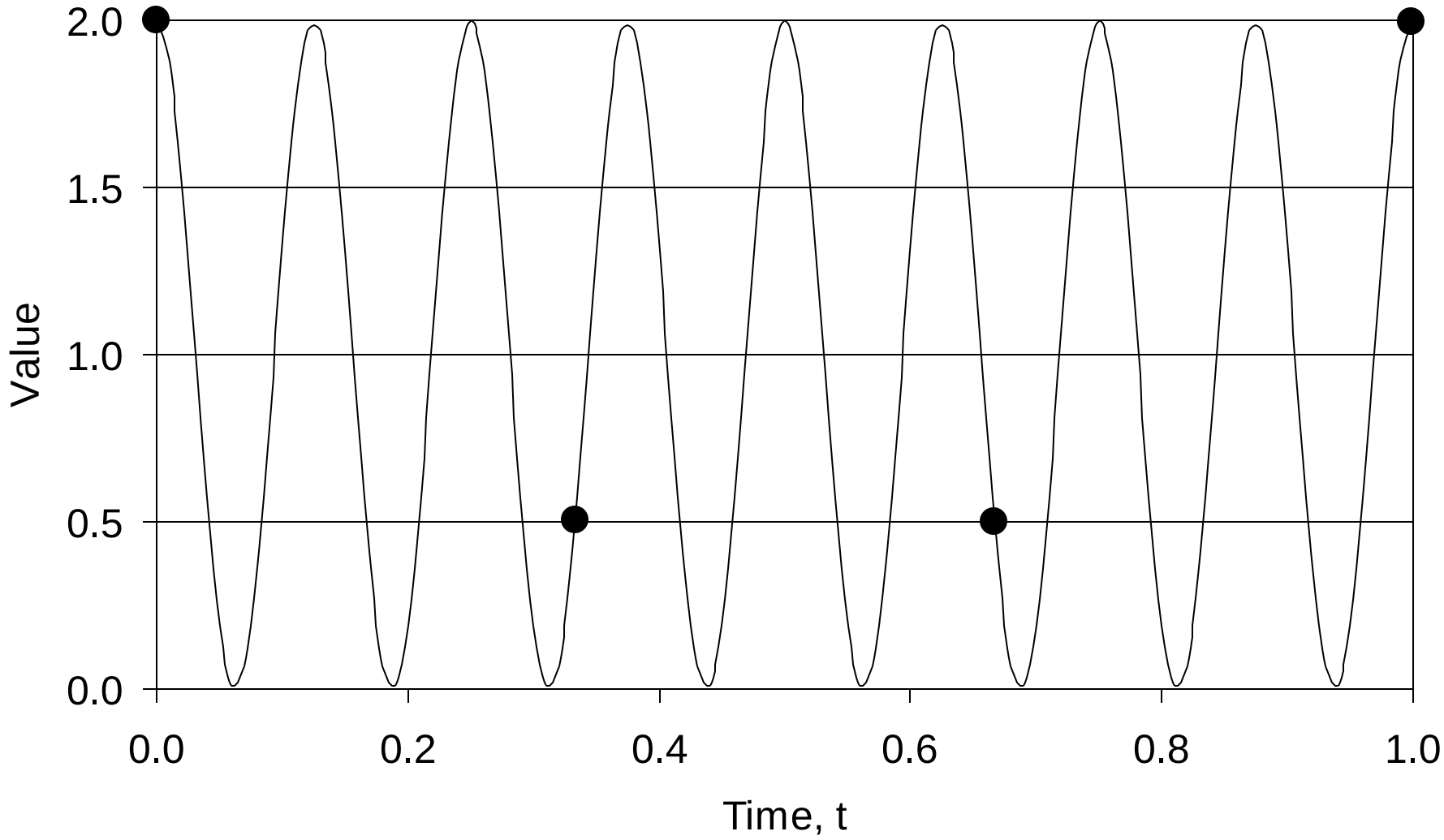
Aliasing

Samples of Different Signals Have the Same Sampled Values



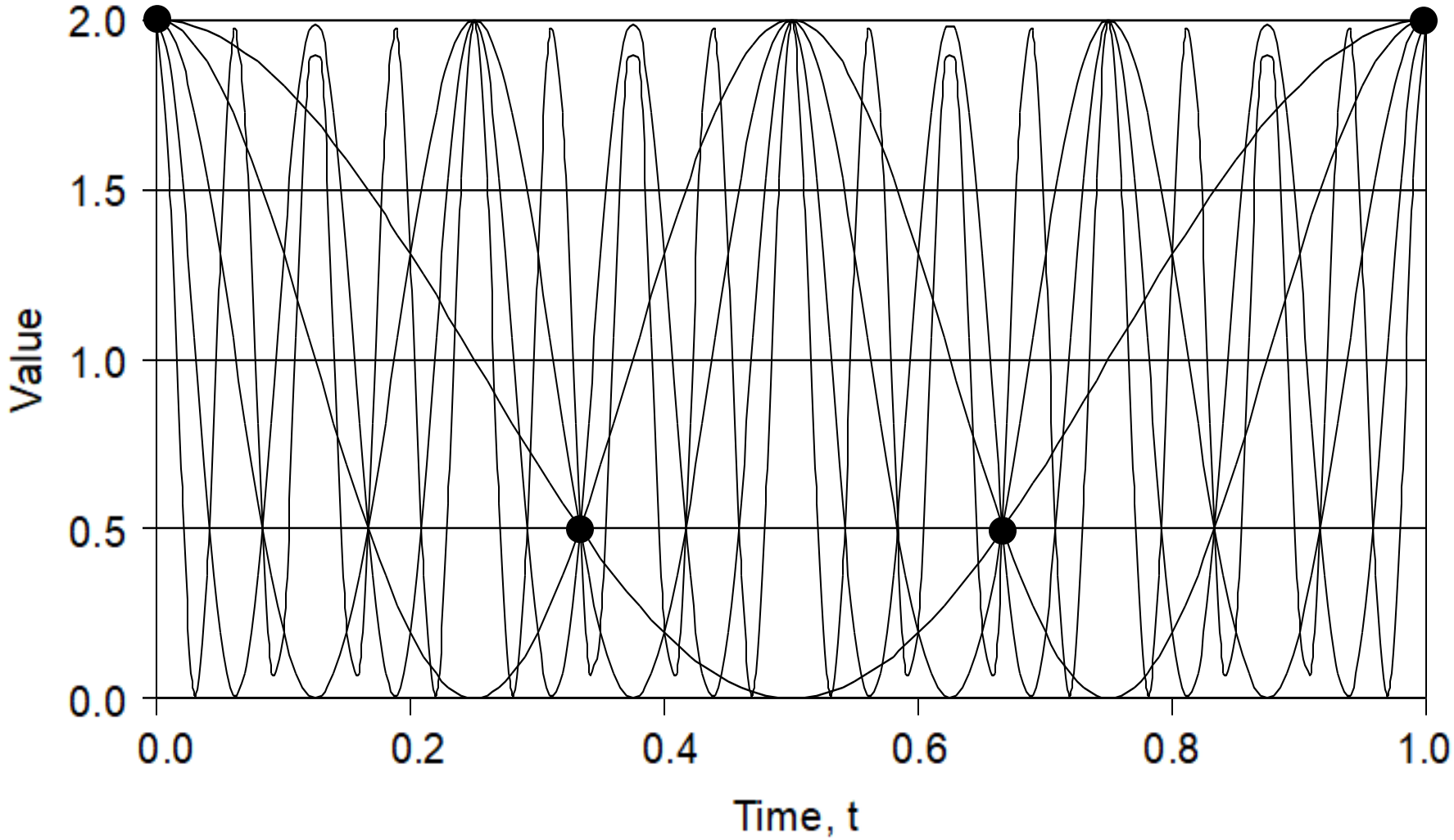
Aliasing

Samples of Different Signals Have the Same Sampled Values



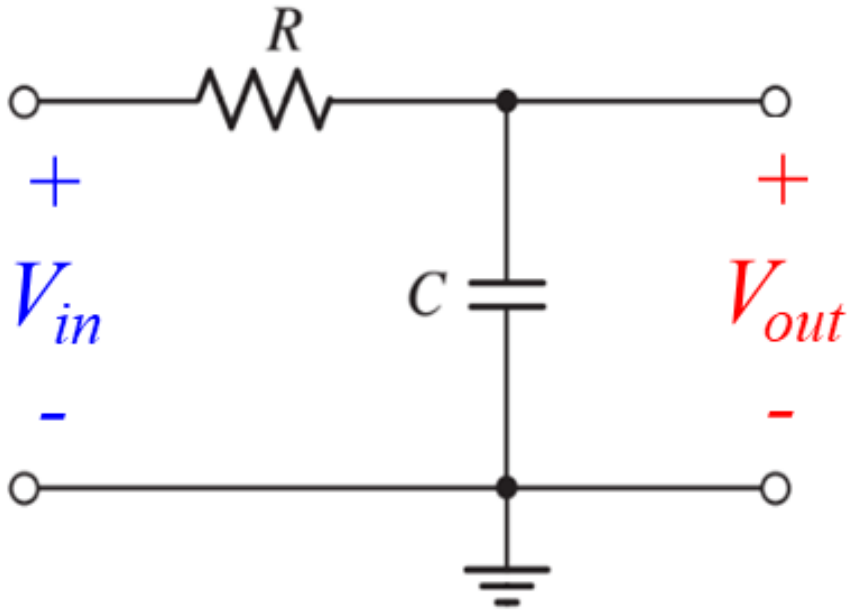
Aliasing

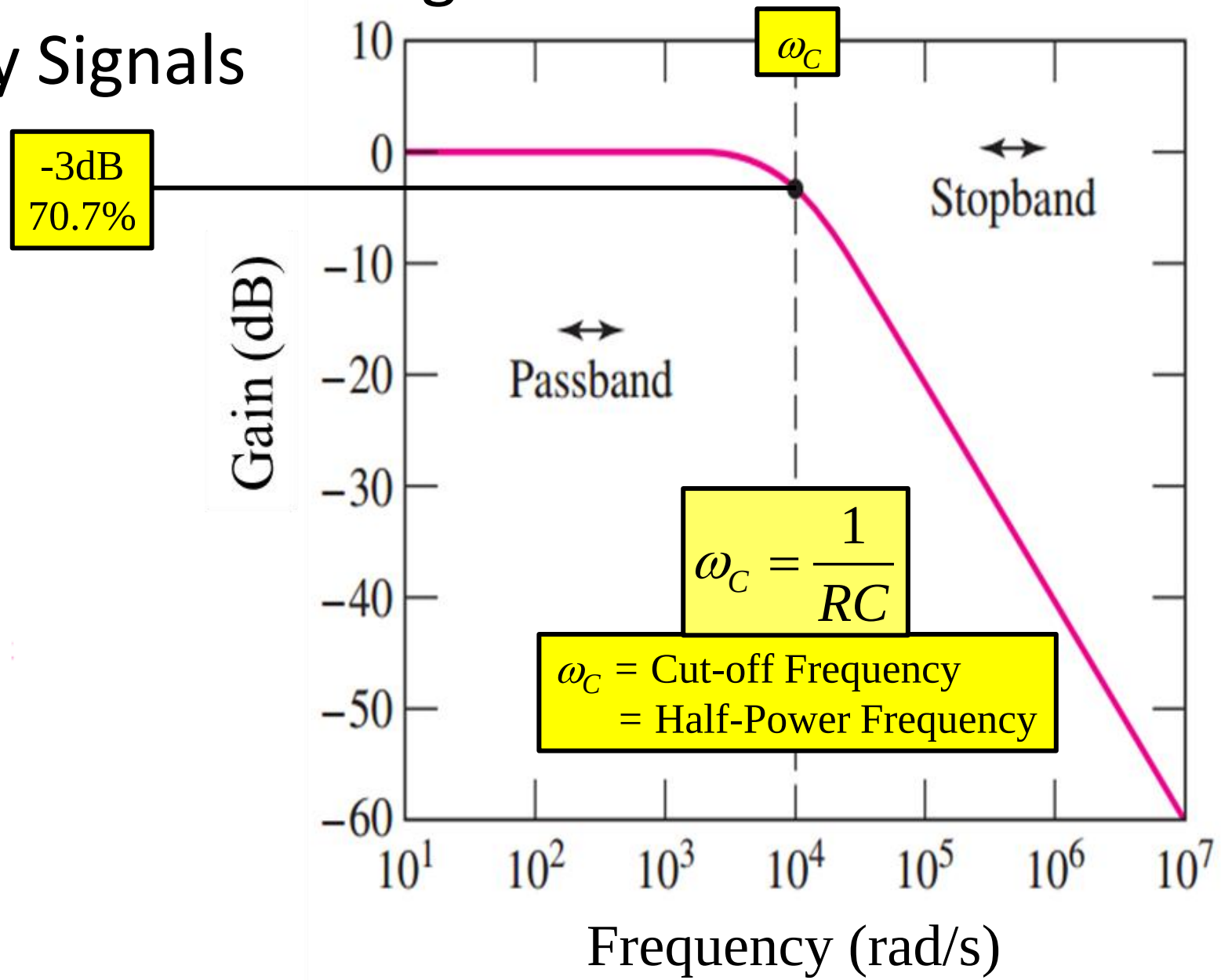
Samples of Different Signals Have the Same Sampled Values



COMING ATTRACTIONS: Anti-Aliasing Filter:
Reduce High Frequency Signals

$$Gain(\text{dB}) = 20\log_{10}\frac{|V_{out}|}{|V_{in}|}$$



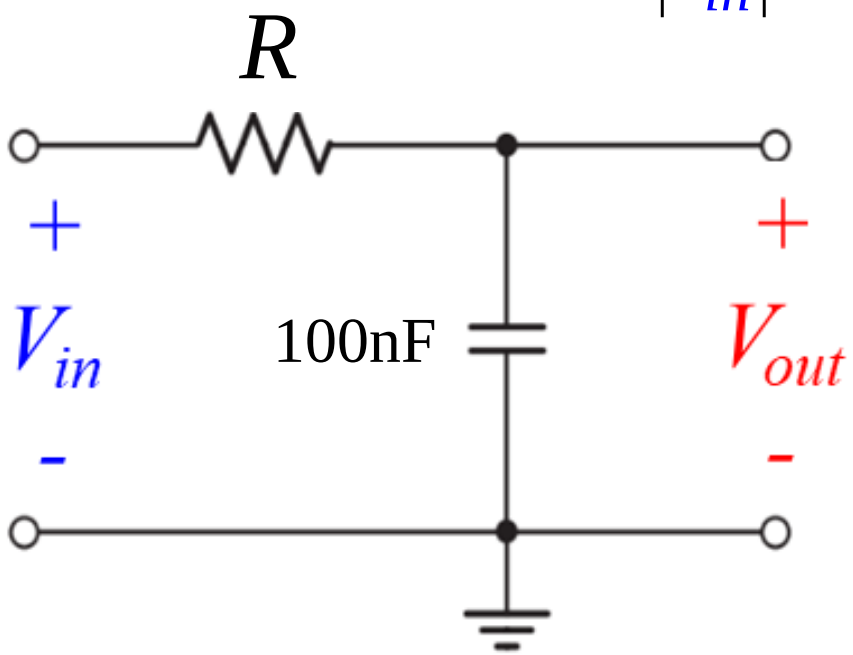


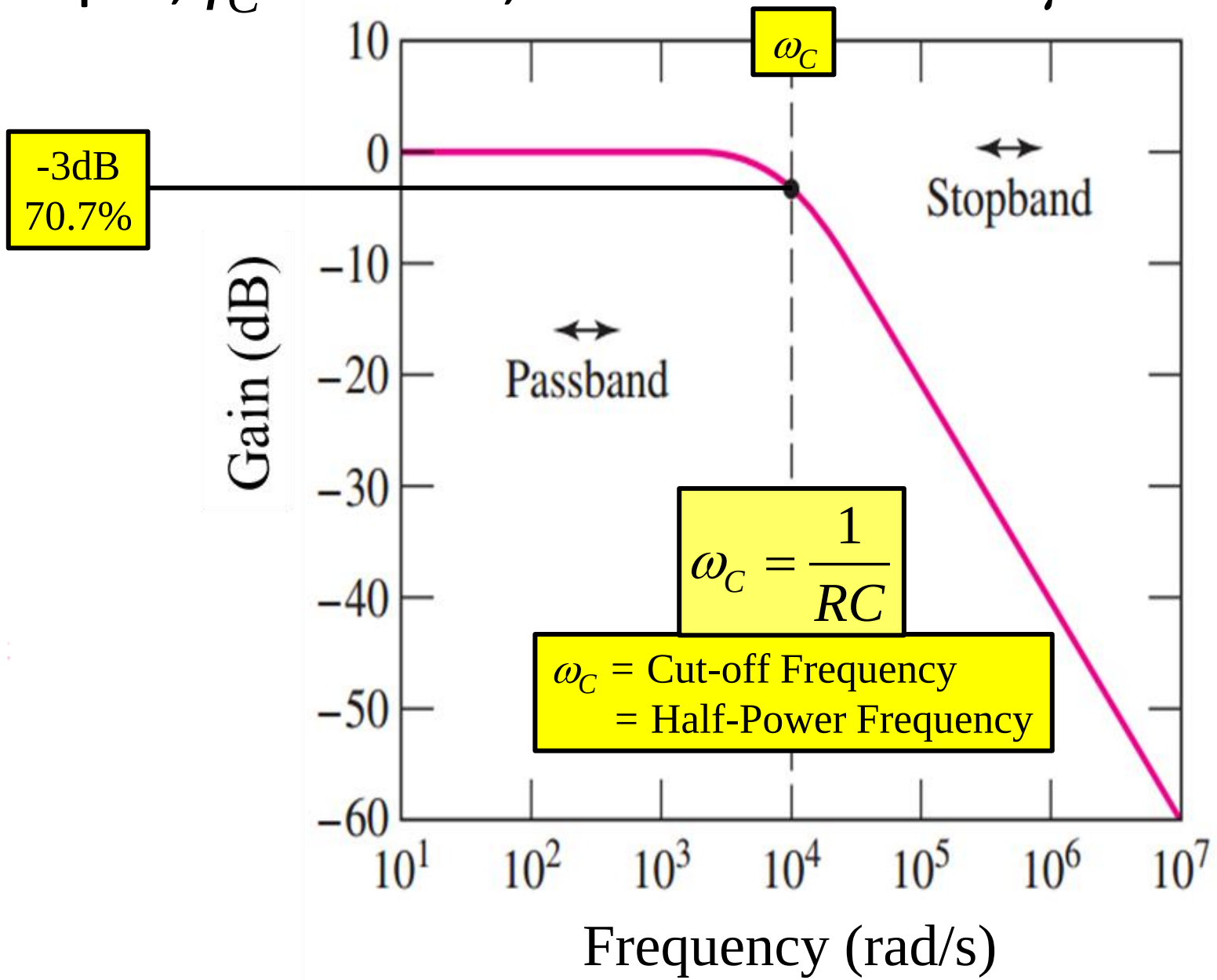
Electrical & Computer ENGINEERING

Anti-Aliasing Filter: Example, $f_C = 30\text{Hz}$, $C = 100\text{nF} = 0.1\mu\text{F}$

Find R

$$Gain(\text{dB}) = 20\log_{10}\frac{|V_{out}|}{|V_{in}|}$$





Do I Really Need to Add a Signal Conditioning Circuit in Front of an ADC Input?

- Short answer: Yes

- Long answer: Yes, but sometimes it's already done for you.
- Many ADCs have a pretty good analog filter built in
  - Integrated filters typical have f_C near 50% of their maximum sampling rate (Nyquist-Shannon)
  - Works well if you are sampling at maximum rate (max power, max busy….)
  - Does not work well if you are sampling below the maximum rate
- An additional filter will only further reduce higher frequency components

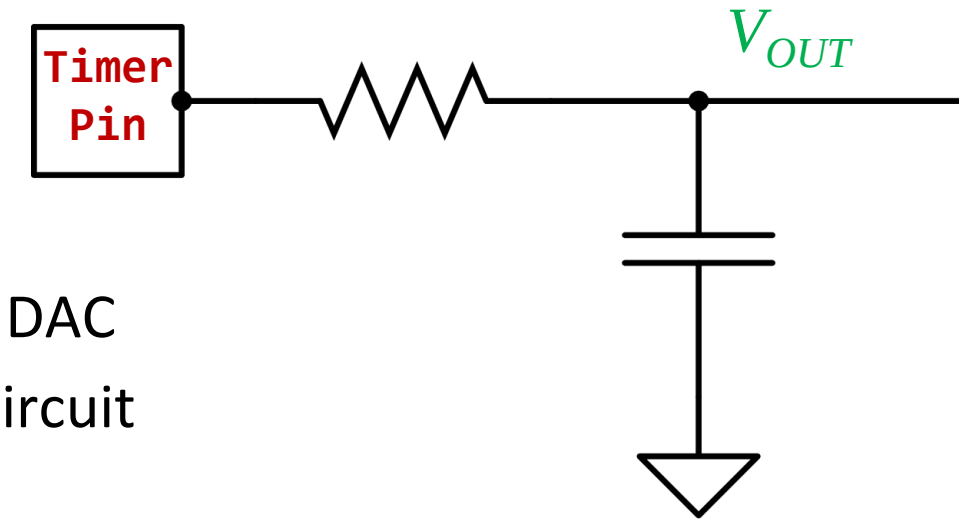
Agenda

- Analog and Digital Worlds – The Problem with Pizza
- Analog to Digital Converters (ADCs)
- Digital to Analog Converters (DACs)

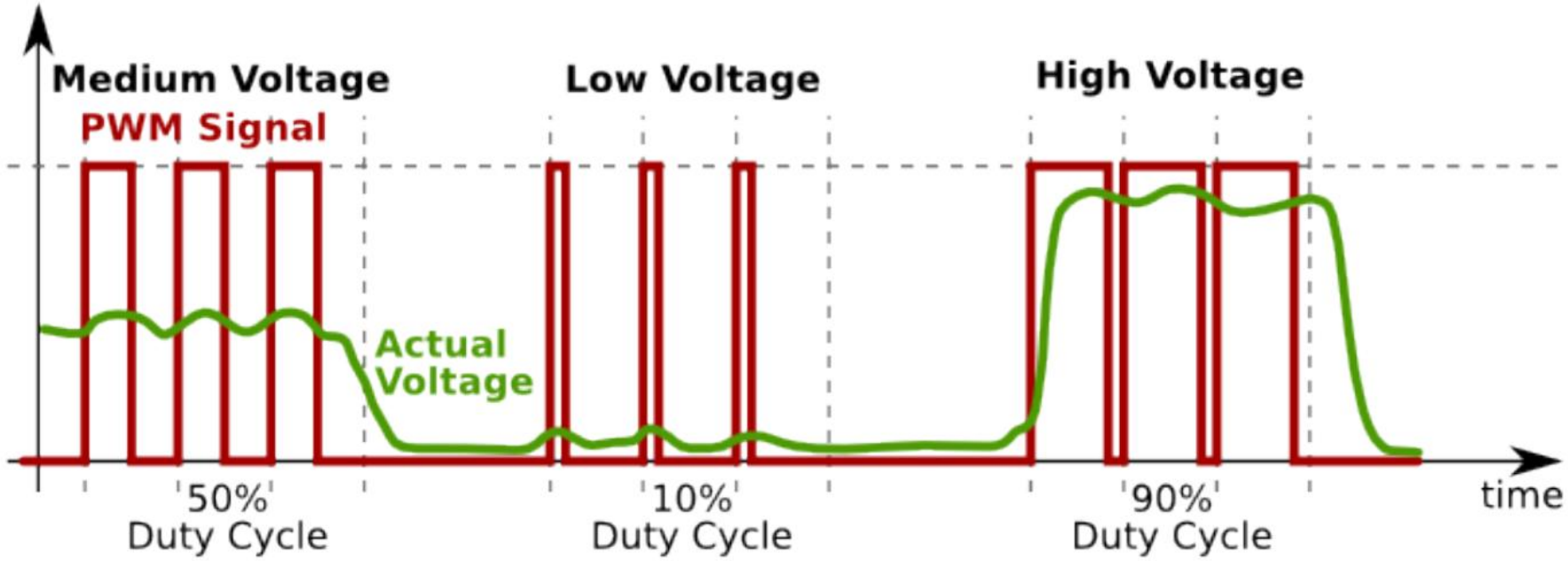
Do It Yourself DAC

- DACs are expensive circuits
- Vary rare for a MCU to have a DAC
- Usually, they are an external circuit you have to purchase

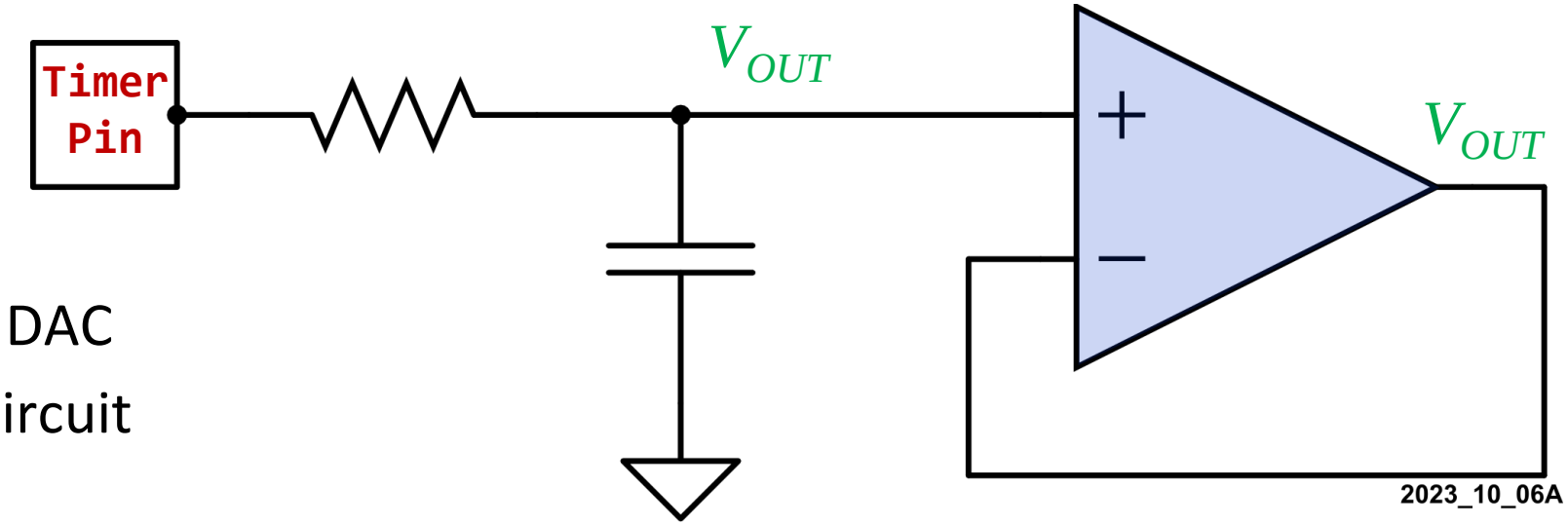
Do It Yourself DAC



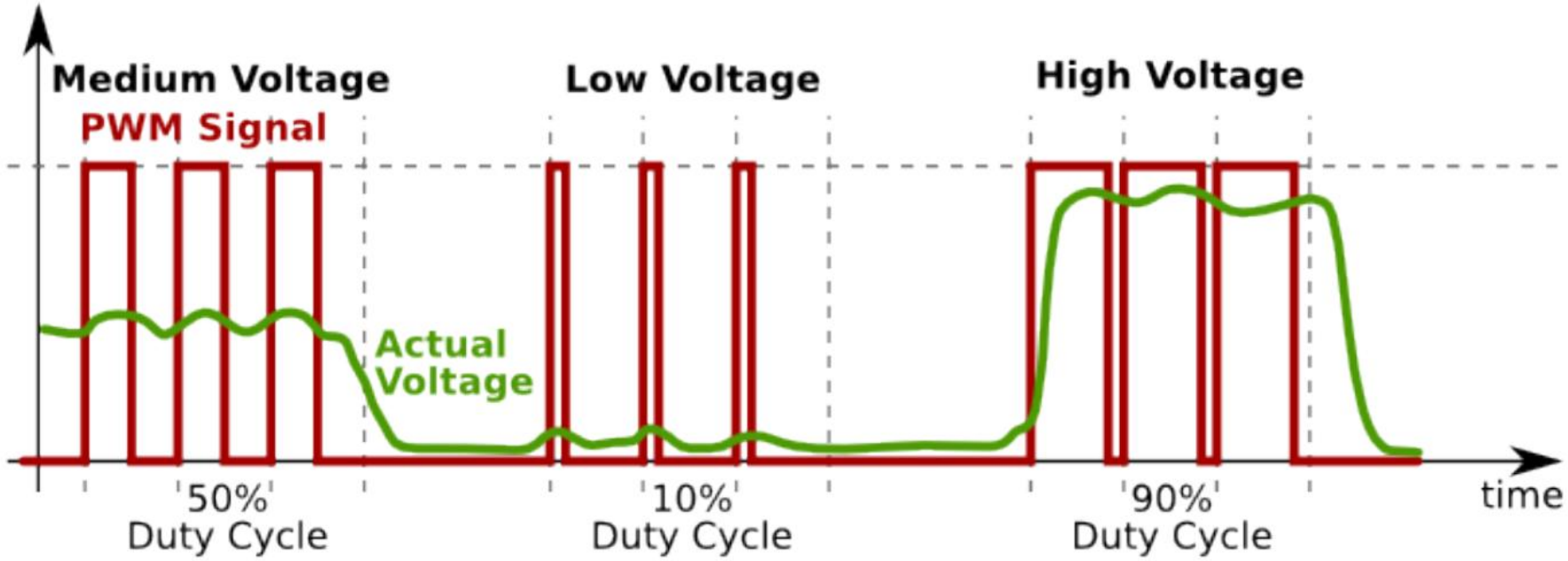
- DACs are expensive circuits
- Vary rare for a MCU to have a DAC
- Usually, they are an external circuit you have to purchase
- You can make a low-quality DAC with just a PWM timer and an RC circuit



Do It Yourself DAC

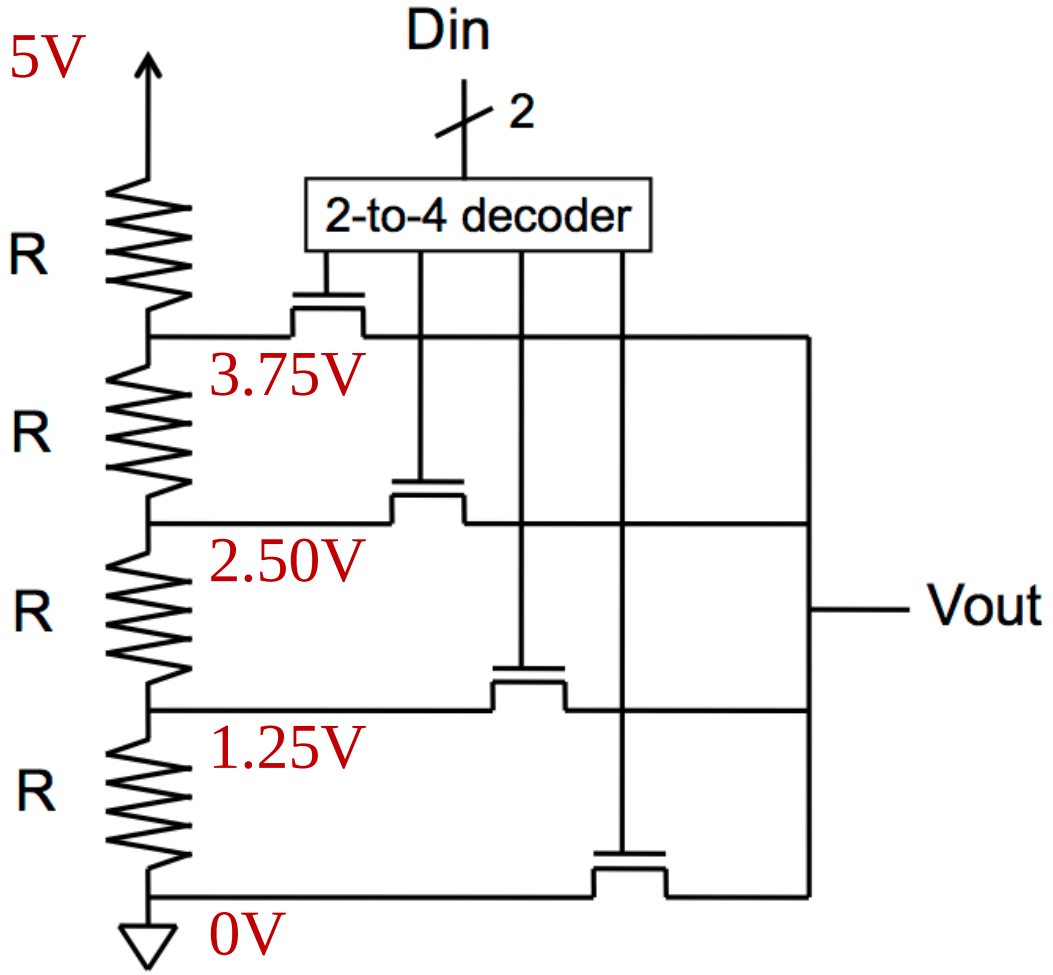


- DACs are expensive circuits
- Vary rare for a MCU to have a DAC
- Usually, they are an external circuit you have to purchase
- You can make a low-quality DAC with just a PWM timer and an RC circuit
- **Use an op amp to increase output current**



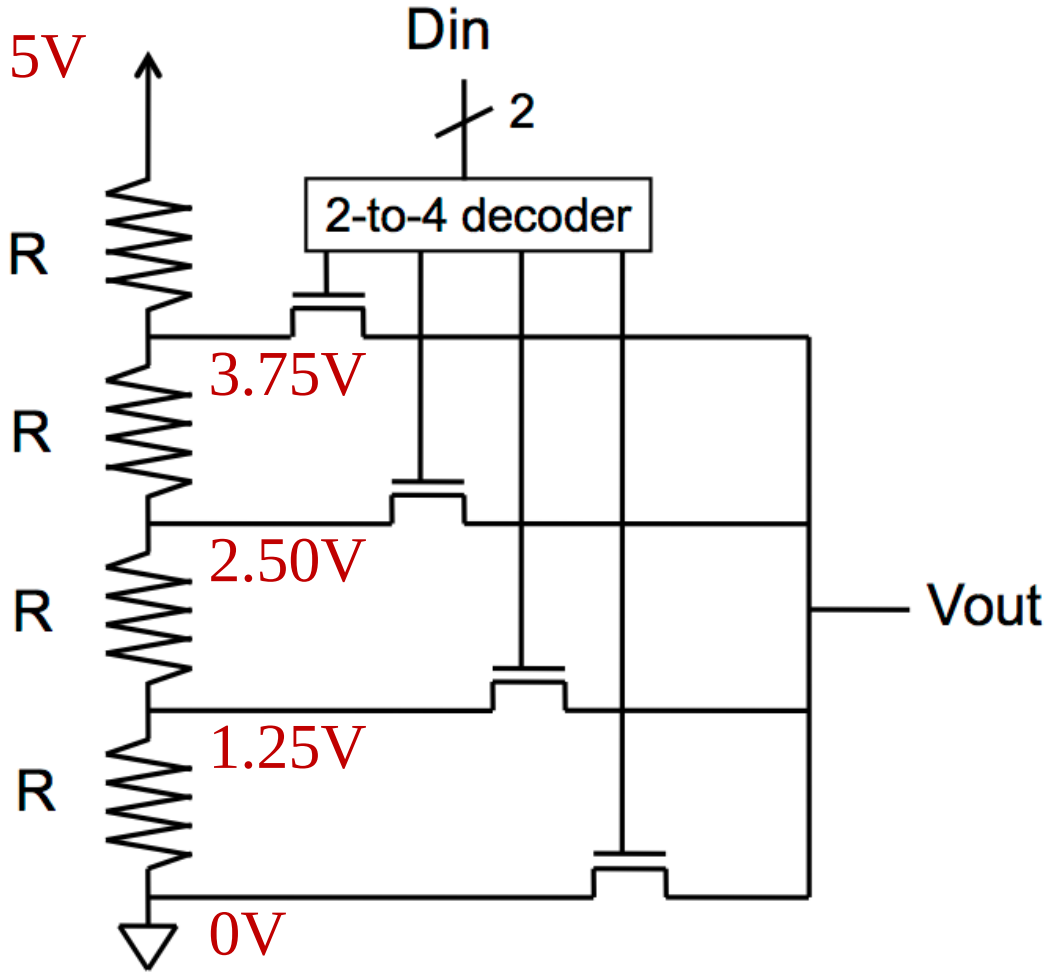
Resistance Ladder DAC

- Advantages
  - Extremely Fast!
  - Monotonic: Output voltage always increases with DAC step count



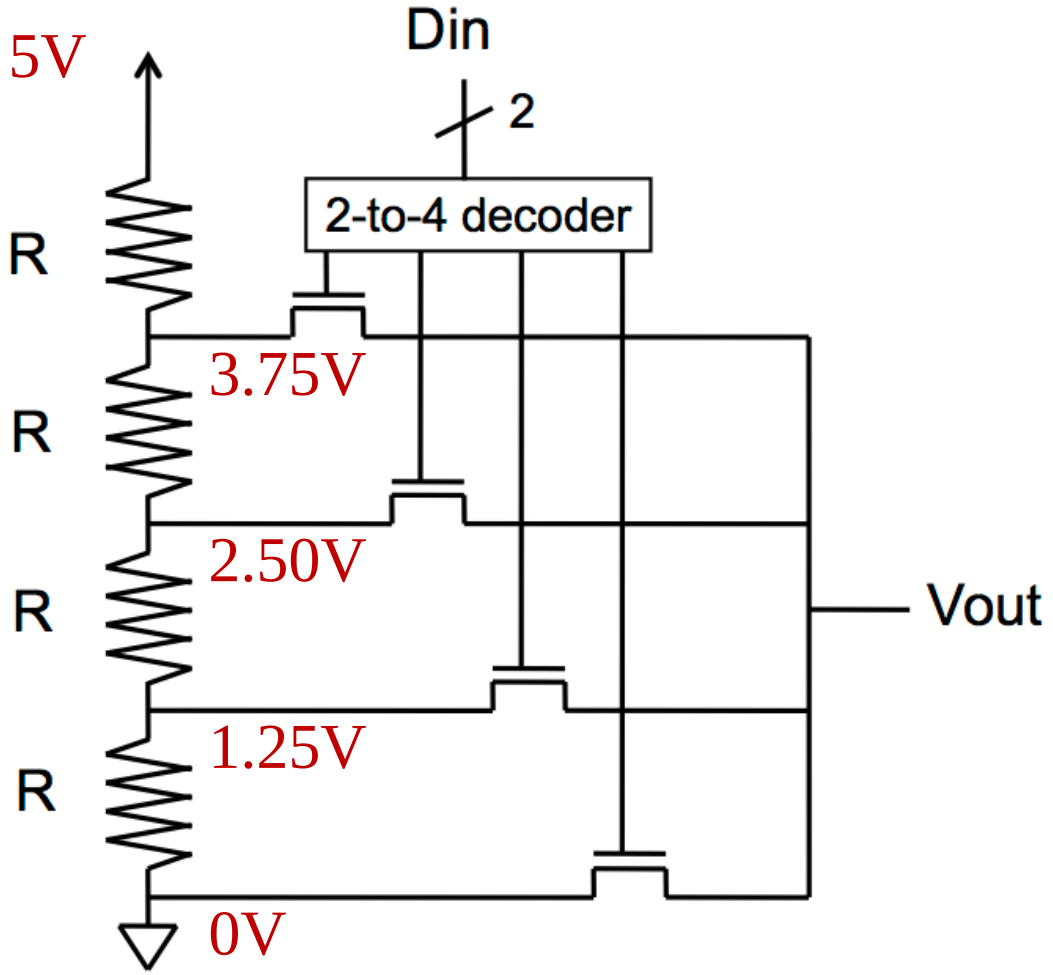
Resistance Ladder DAC

- Advantages
  - Extremely Fast!
  - Monotonic: Output voltage always increases with DAC step count
- Disadvantages
  - One resistor and one transistor required per each output voltage level (N-bit DAC requires 2^N resistors and transistors)
  - Large decoder needed $\log_2$N-to-N
  - Each extra bit doubles the size



Resistance Ladder DAC

- Advantages
  - Extremely Fast!
  - Monotonic: Output voltage always increases with DAC step count
- Disadvantages
  - One resistor and one transistor required per each output voltage level (N-bit DAC requires 2^N resistors and transistors)
  - Large decoder needed $\log_2$N-to-N
  - Each extra bit doubles the size
  - Accuracy requires matched resistors
  - Use an op amp buffer to increase current drive

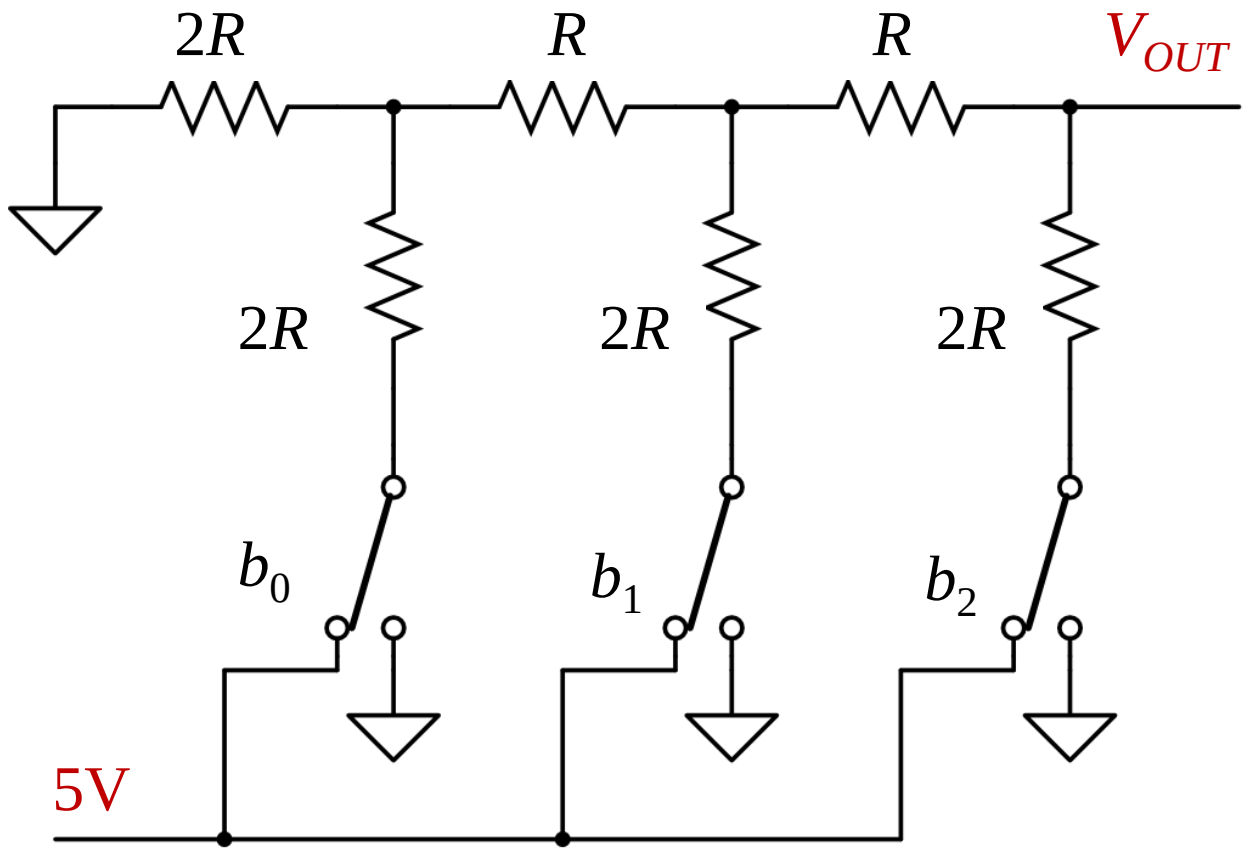


R-$2R$ Ladder DAC

- Resistive ladder constructed from R and $2R$ resistor values
- Switch positions controlled by the DAC's N-bits of inputs (3-bit DAC shown)

- b_X = **1** Switch to +5V
- b_X = **0** Switch to 0V

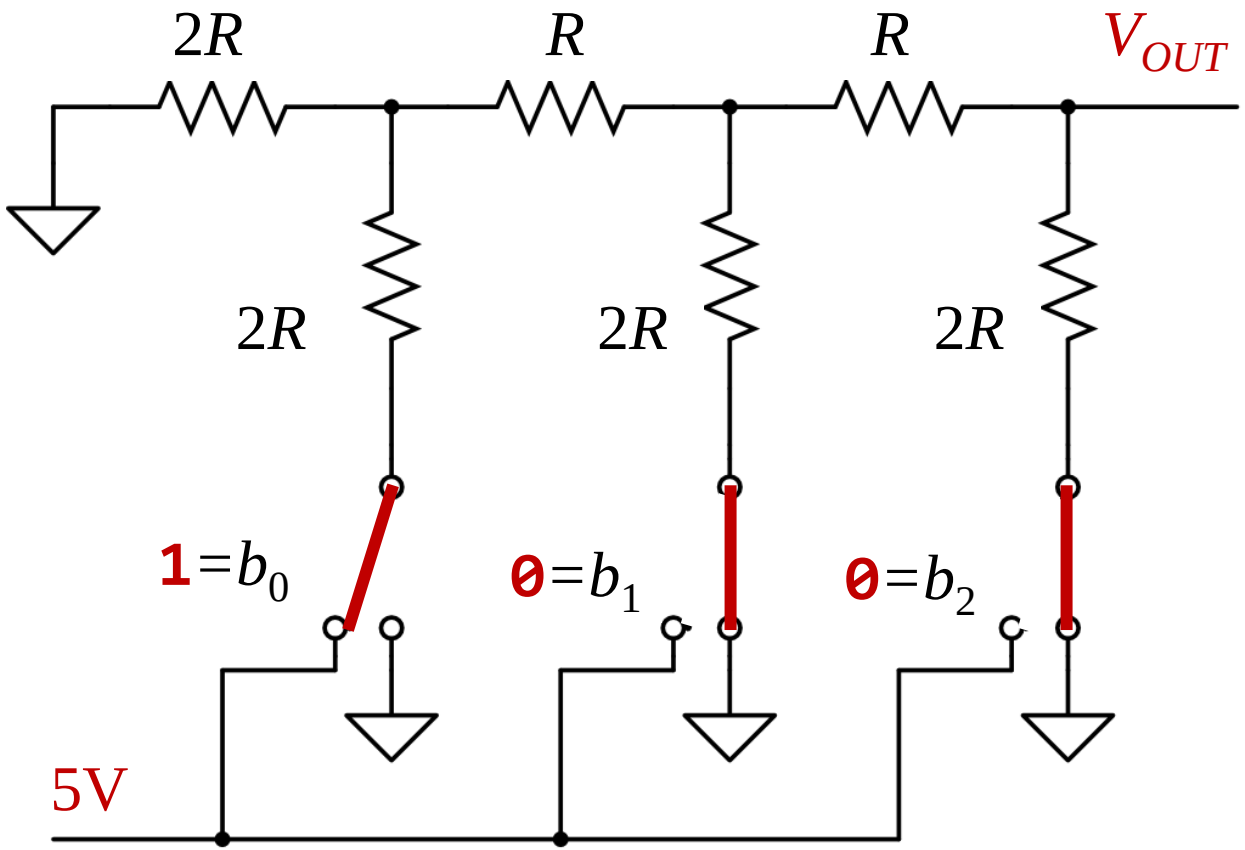
- 3-bit DAC shown has input **111**



2023_10_06B

R-$2R$ Ladder DAC

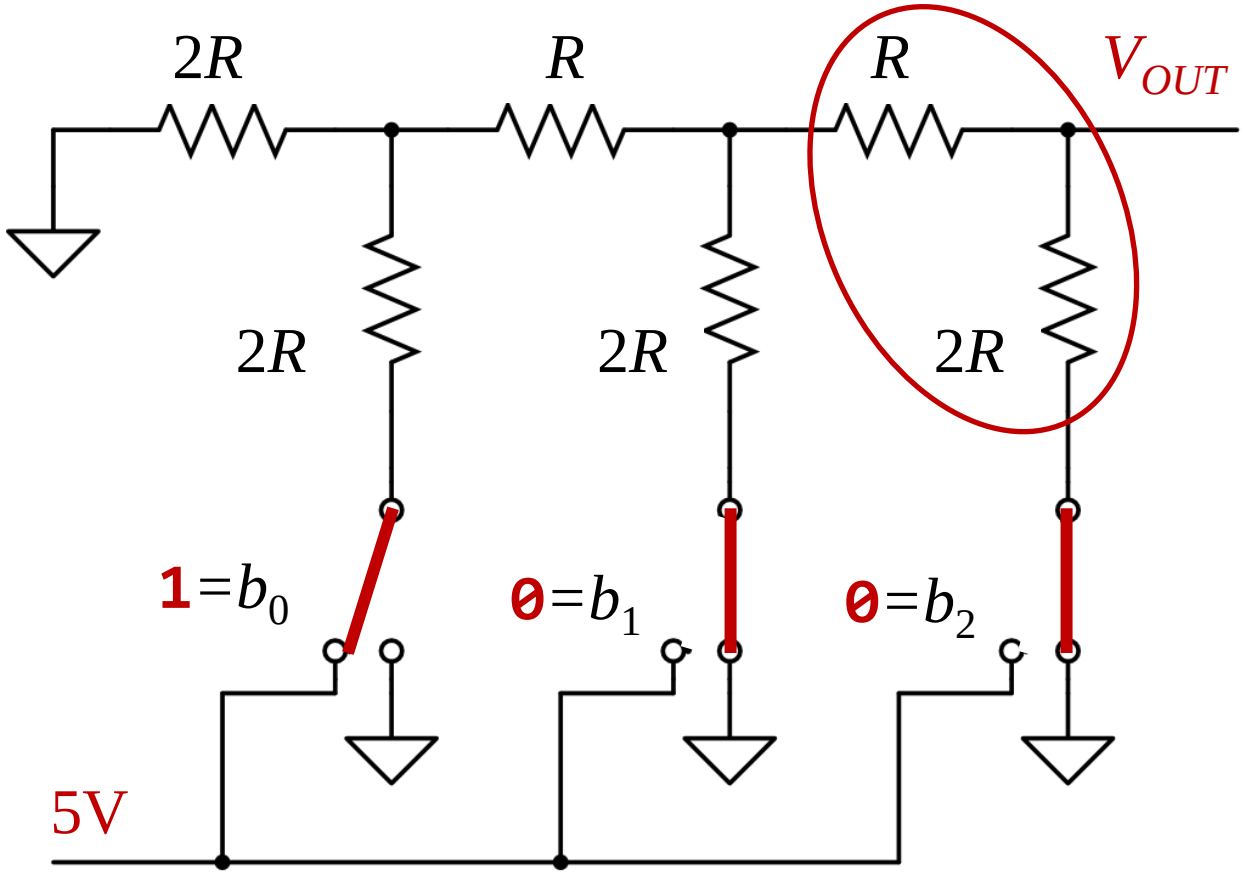
- Consider input **001**



2023_10_06B

R-$2R$ Ladder DAC

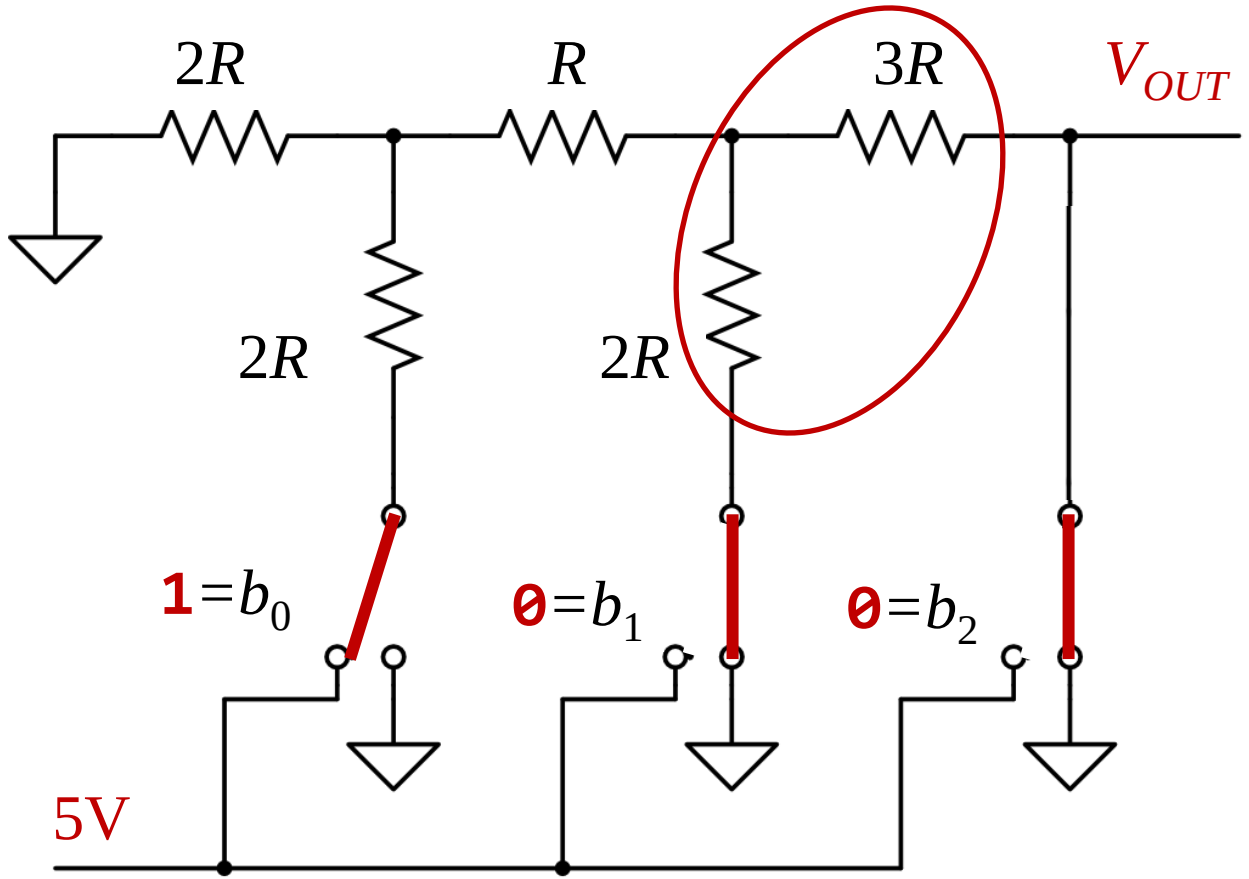
- Consider input **001**



2023_10_06B

R-$2R$ Ladder DAC

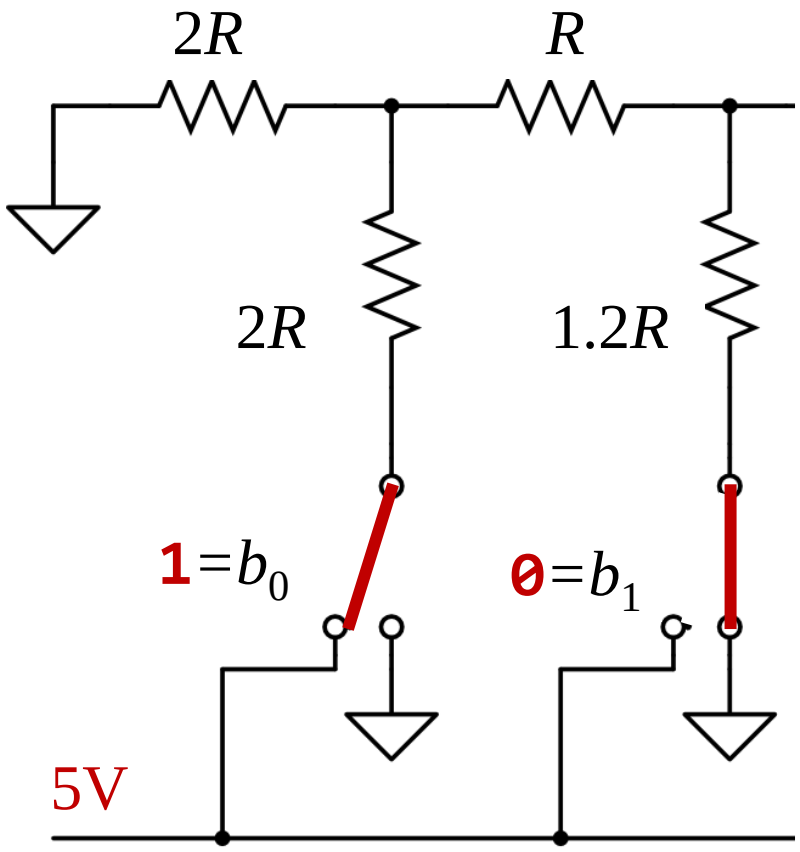
- Consider input **001**



2023_10_06B

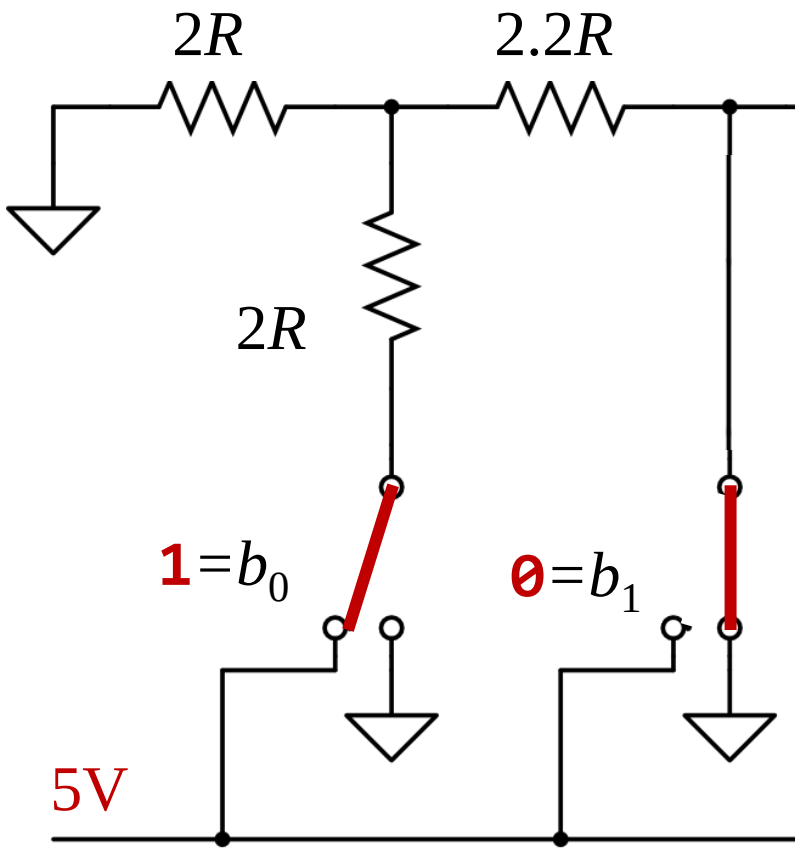
R-$2R$ Ladder DAC

- Consider input **001**



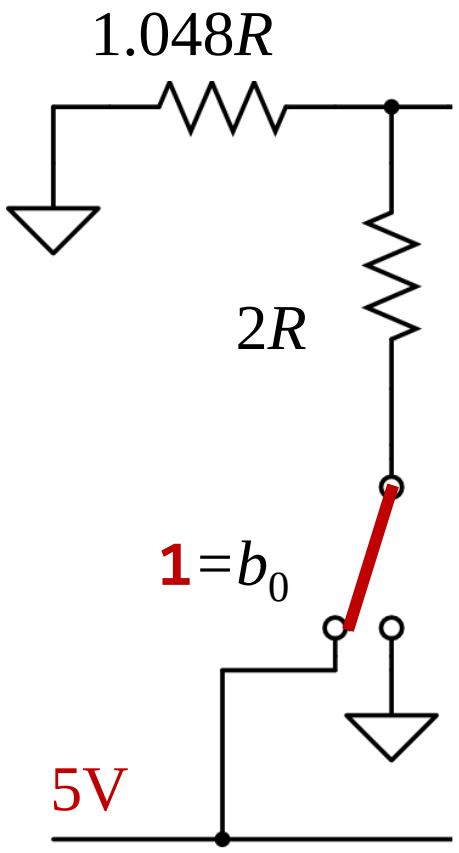
R-$2R$ Ladder DAC

- Consider input **001**



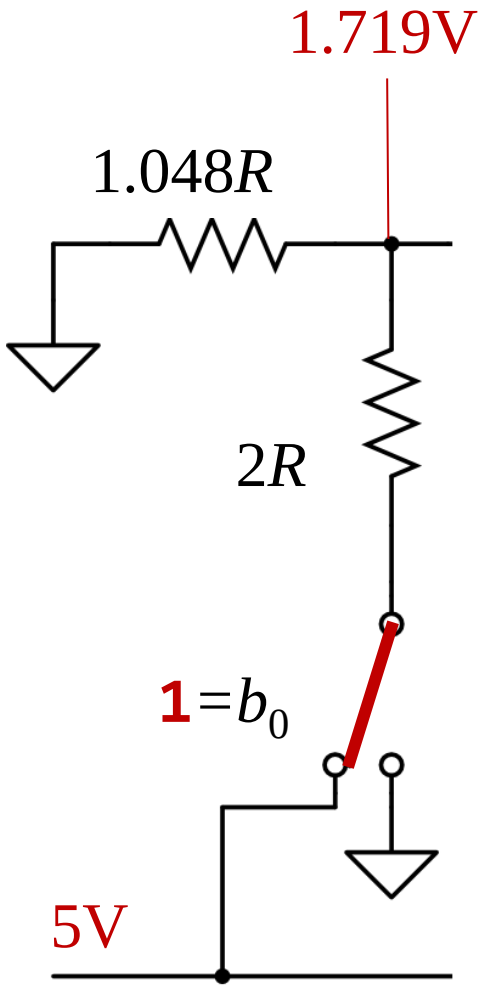
R-$2R$ Ladder DAC

- Consider input **001**



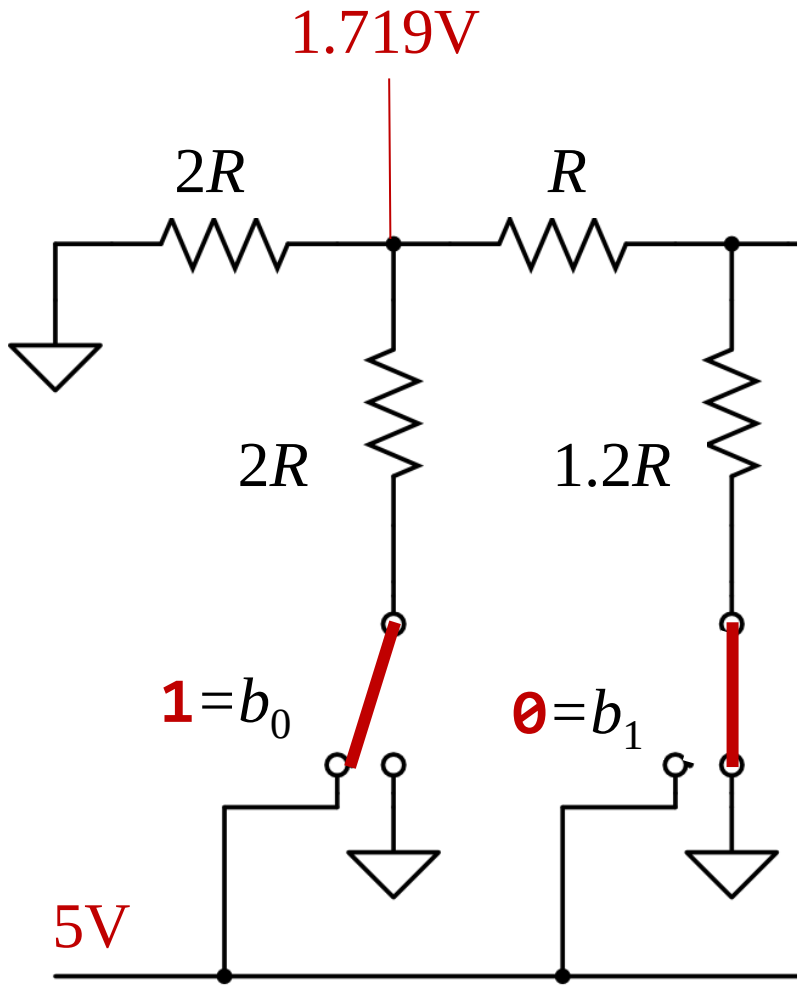
R-$2R$ Ladder DAC

- Consider input **001**



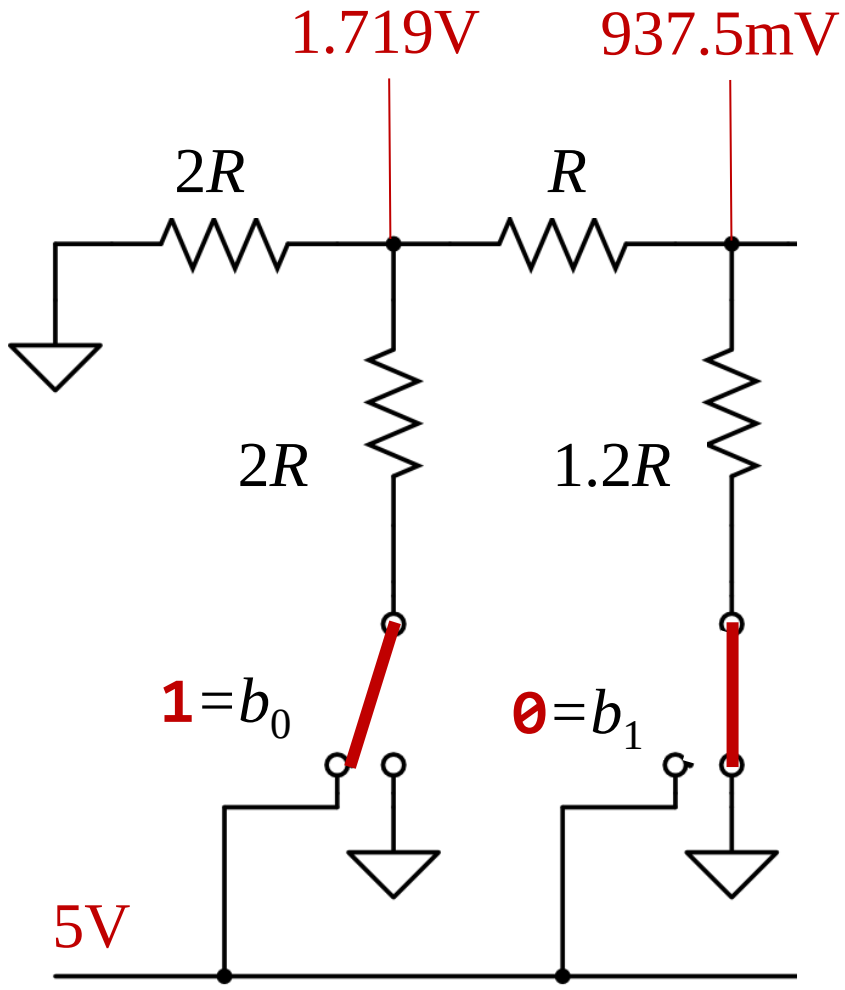
R-$2R$ Ladder DAC

- Consider input **001**



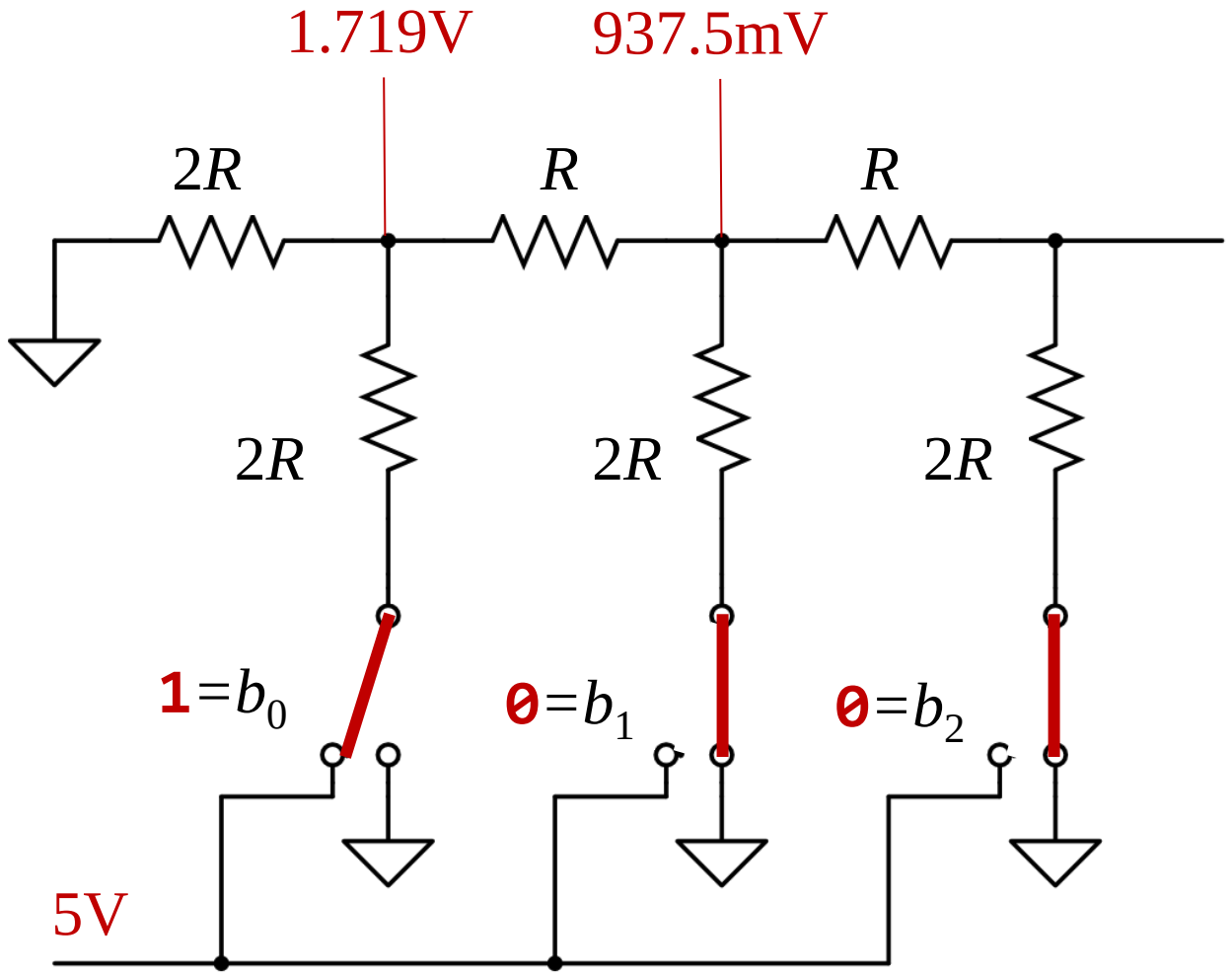
R-$2R$ Ladder DAC

- Consider input **001**



R-$2R$ Ladder DAC

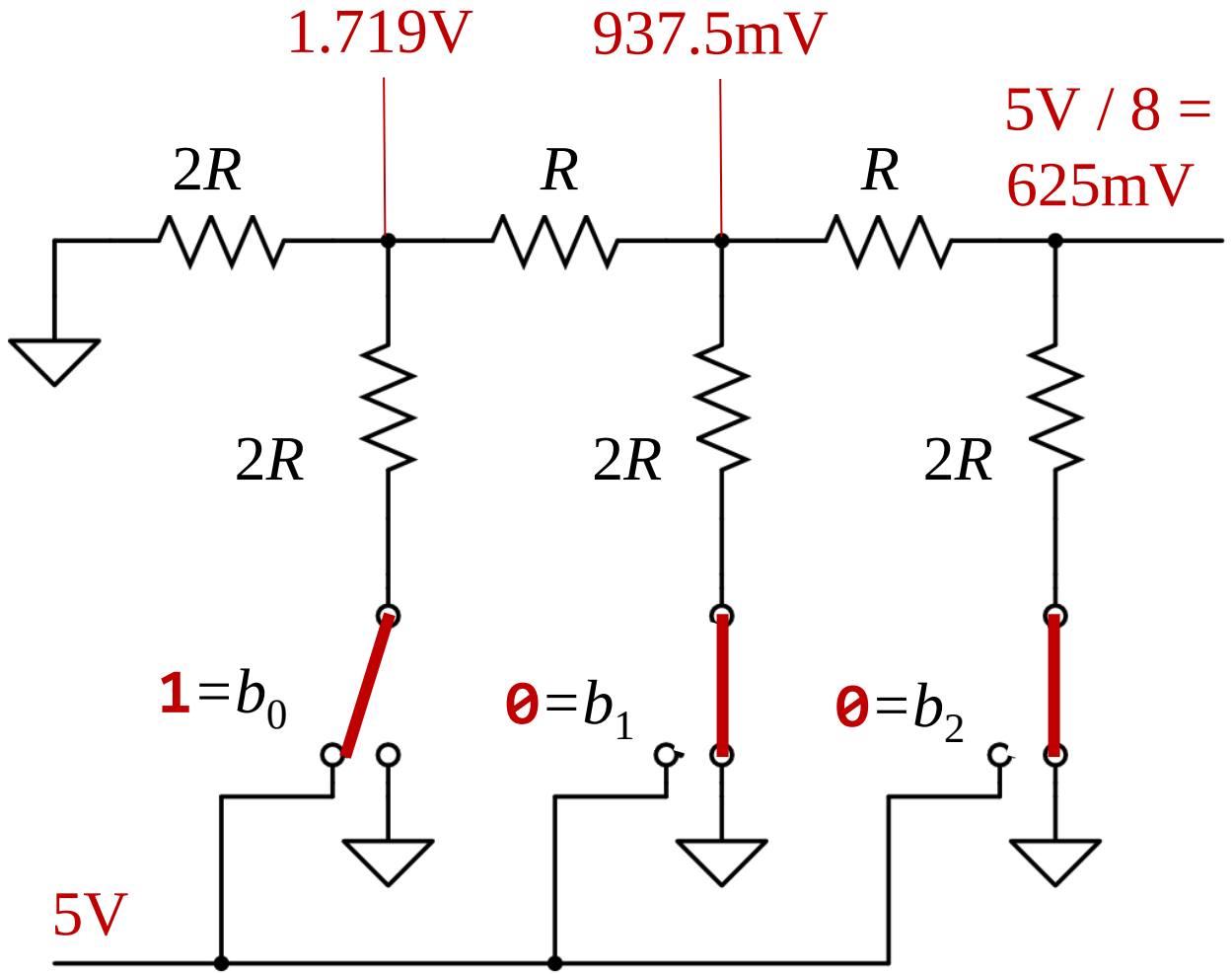
- Consider input **001**



2023_10_06B

R-$2R$ Ladder DAC

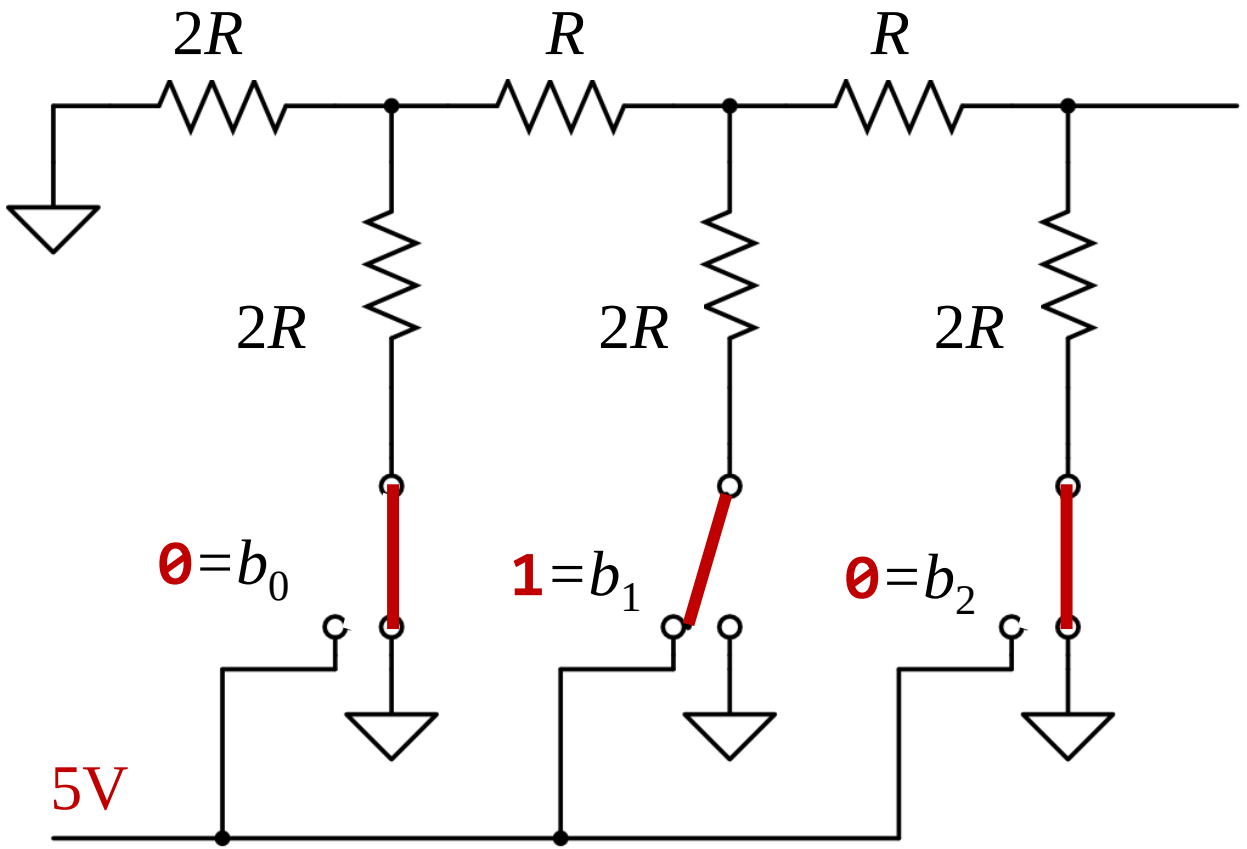
- Consider input **001**



2023_10_06B

R-$2R$ Ladder DAC

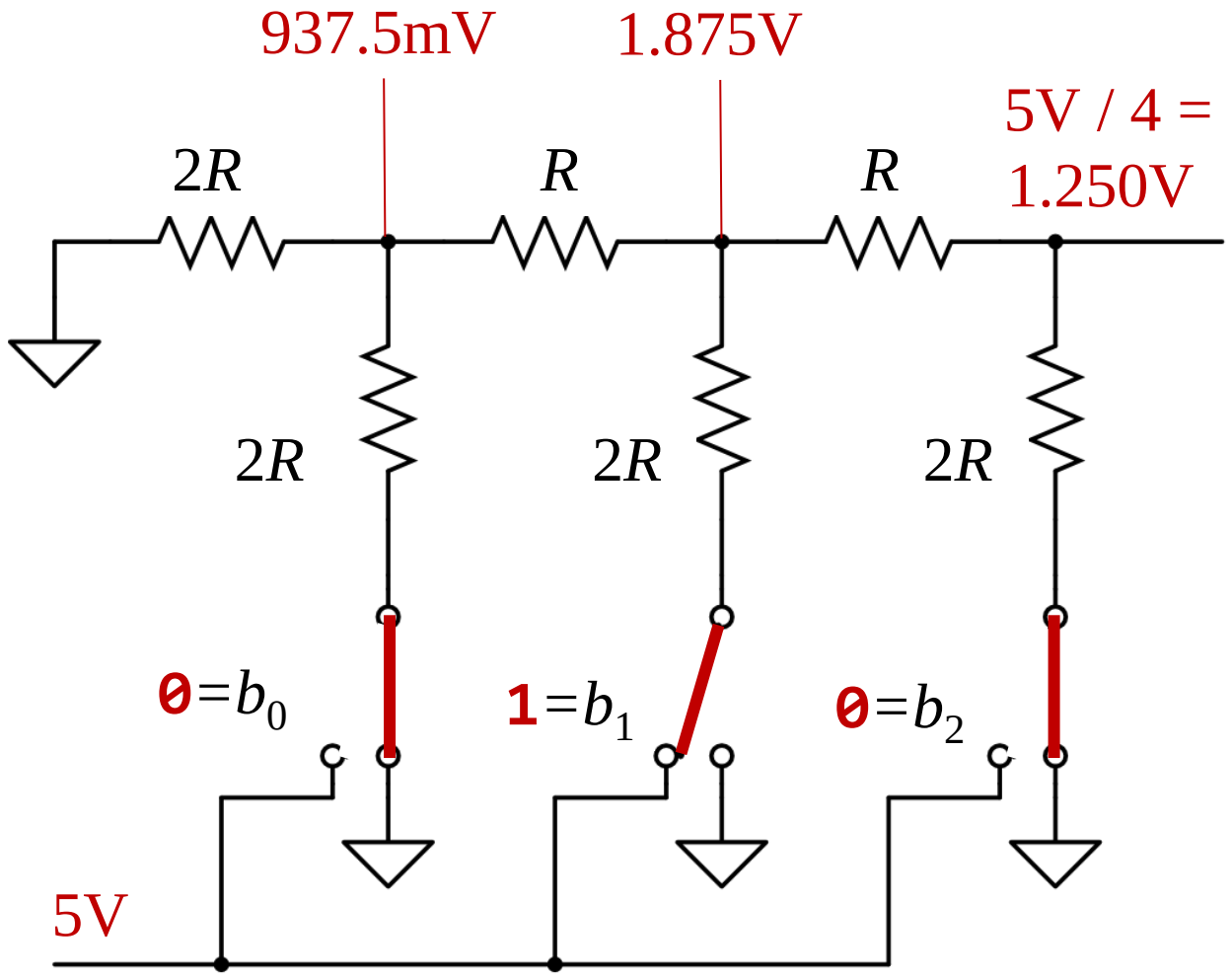
- Consider input **010**



2023_10_06B

R-$2R$ Ladder DAC

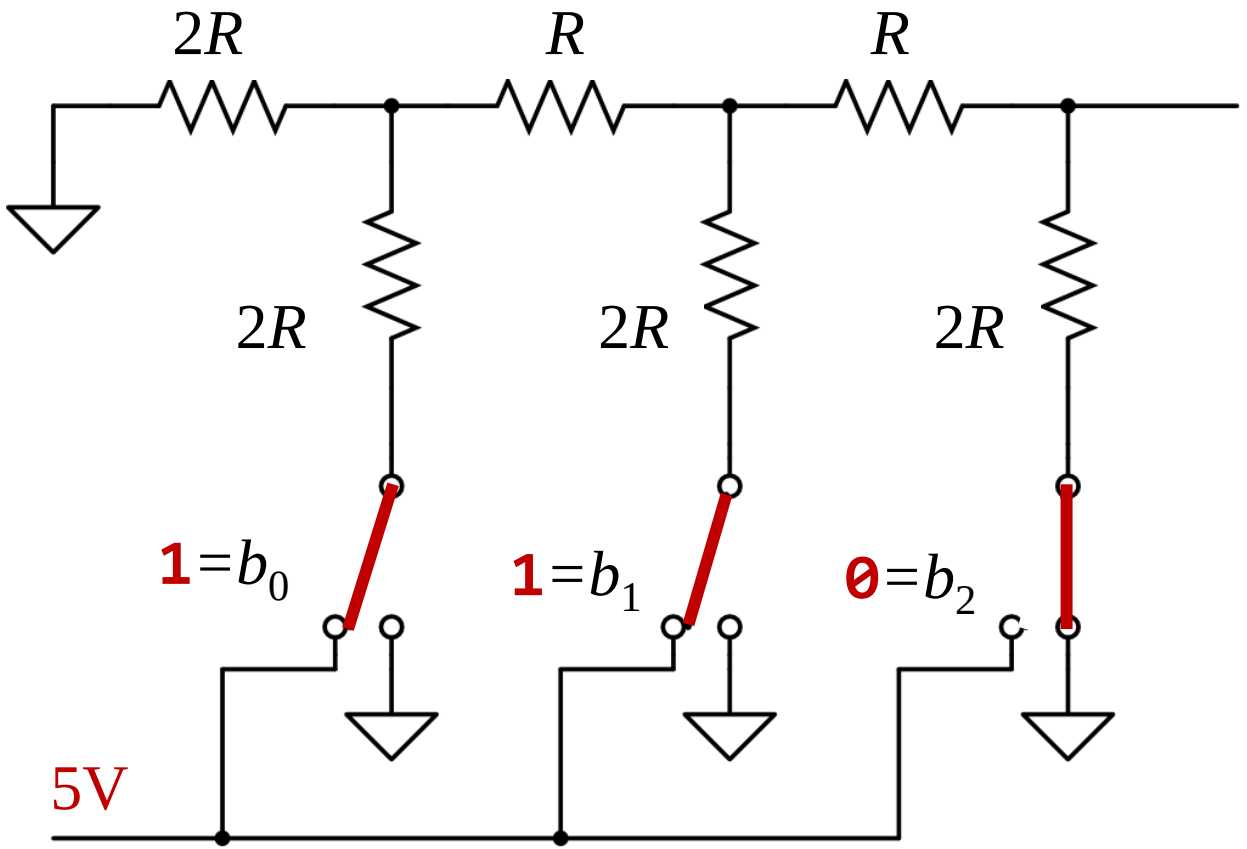
- Consider input **010**



2023_10_06B

R-$2R$ Ladder DAC

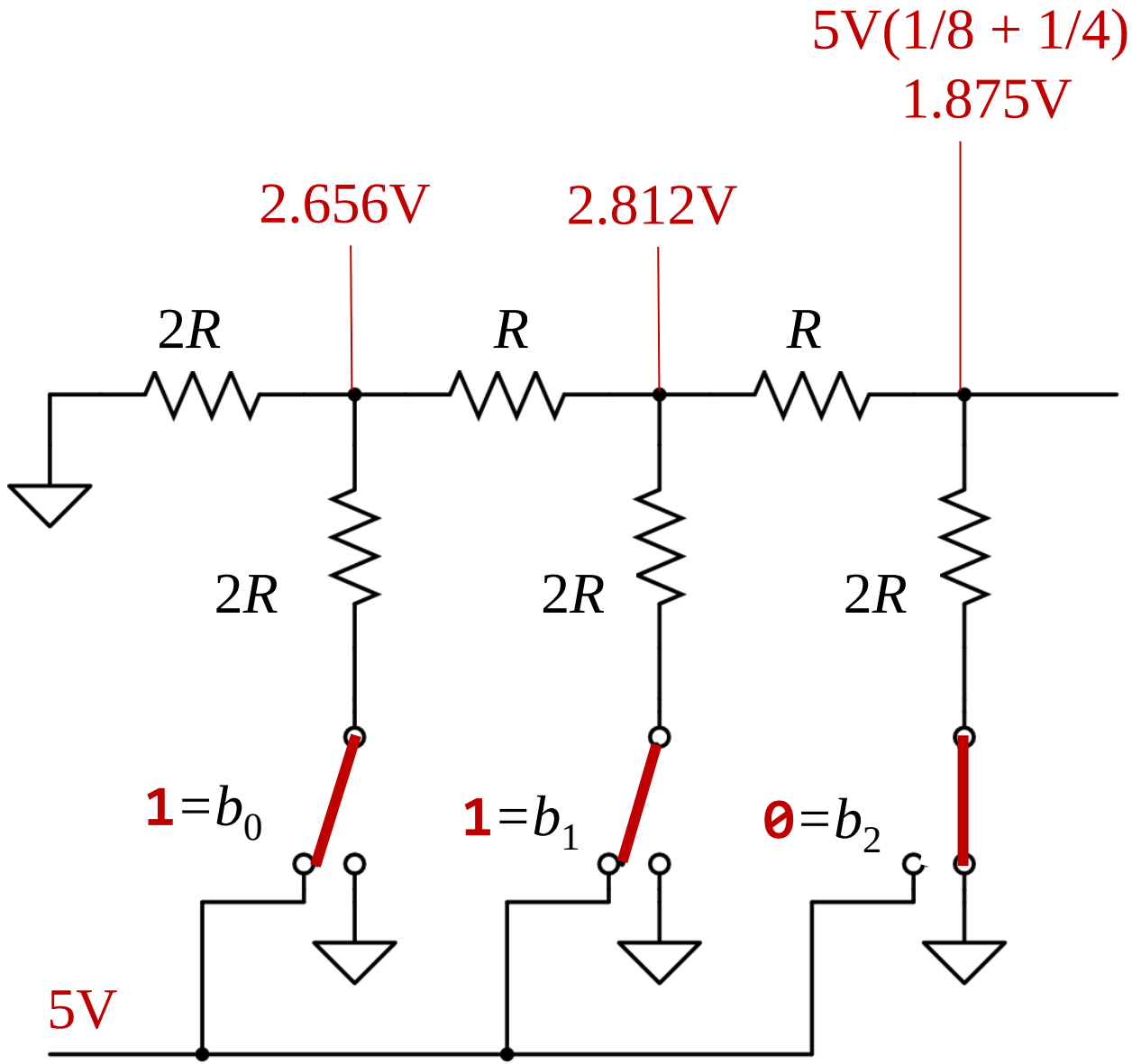
- Consider input **011**



2023_10_06B

R-$2R$ Ladder DAC

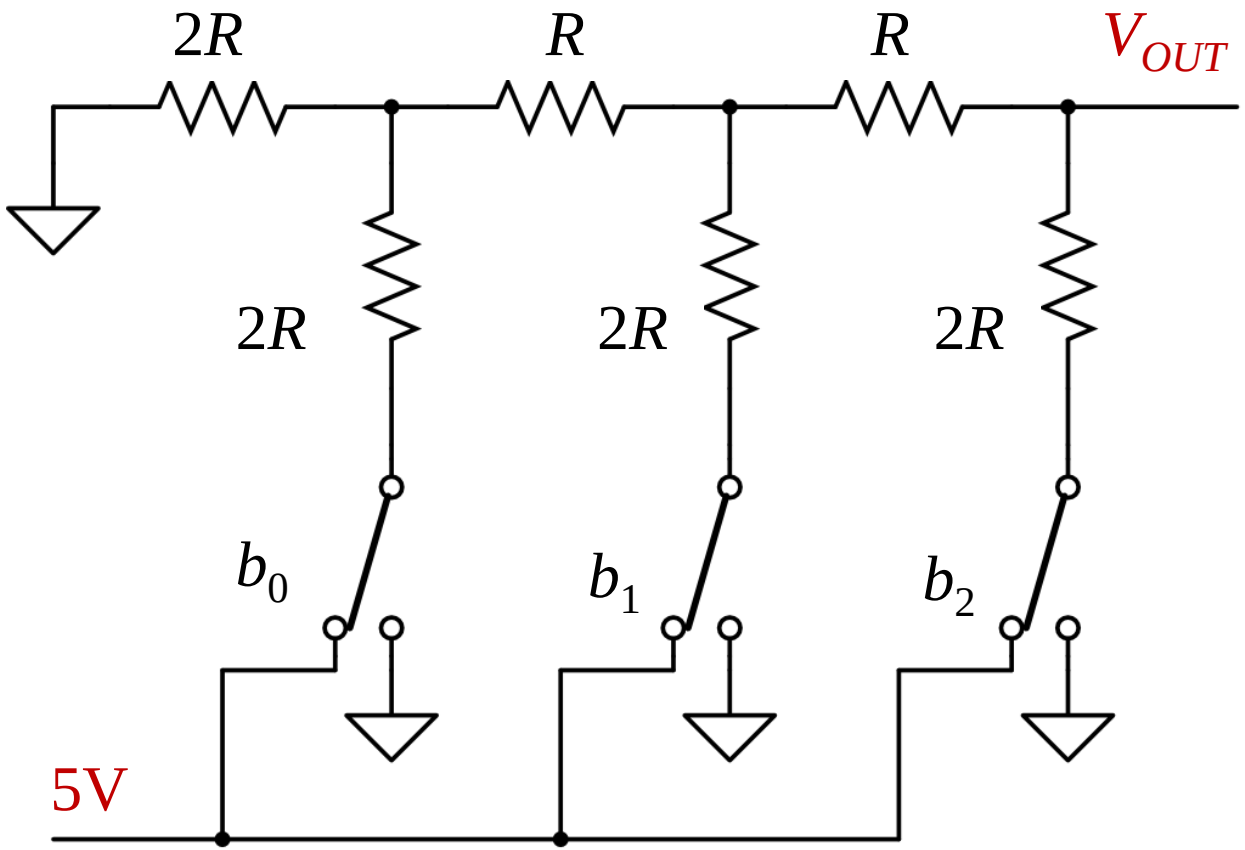
- Consider input **011**



2023_10_06B

R-$2R$ Ladder DAC

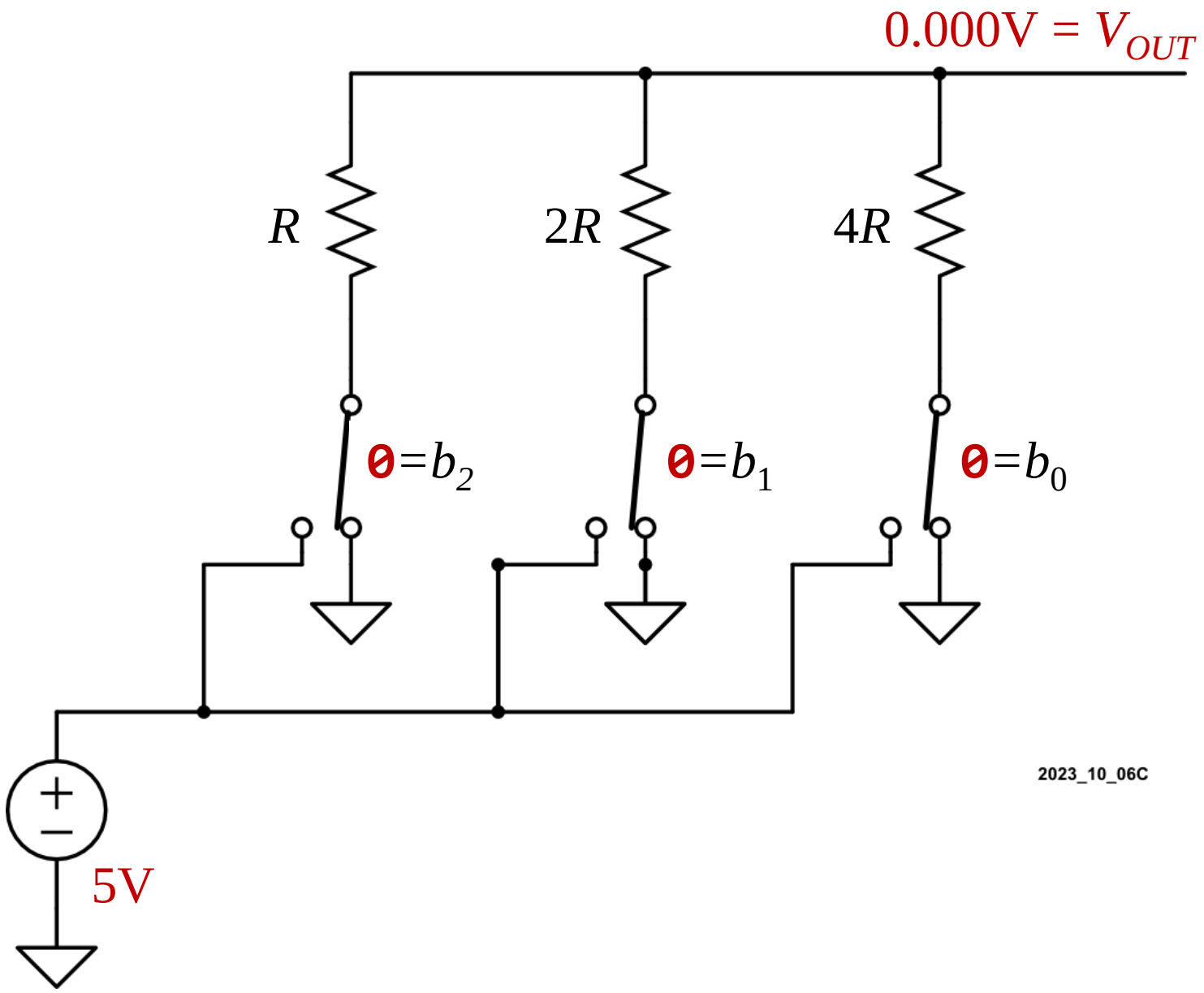
- Advantages
  - Fast!
  - Easy to manufacture for low-bit counts
- Disadvantages
  - Monotonicity problems
  - If resistors aren't completely accurate, increasing the input across 2^X boundaries can decrease the output voltage (01111111 → 10000000)
  - Use an op amp buffer to increase current drive



2023_10_06B

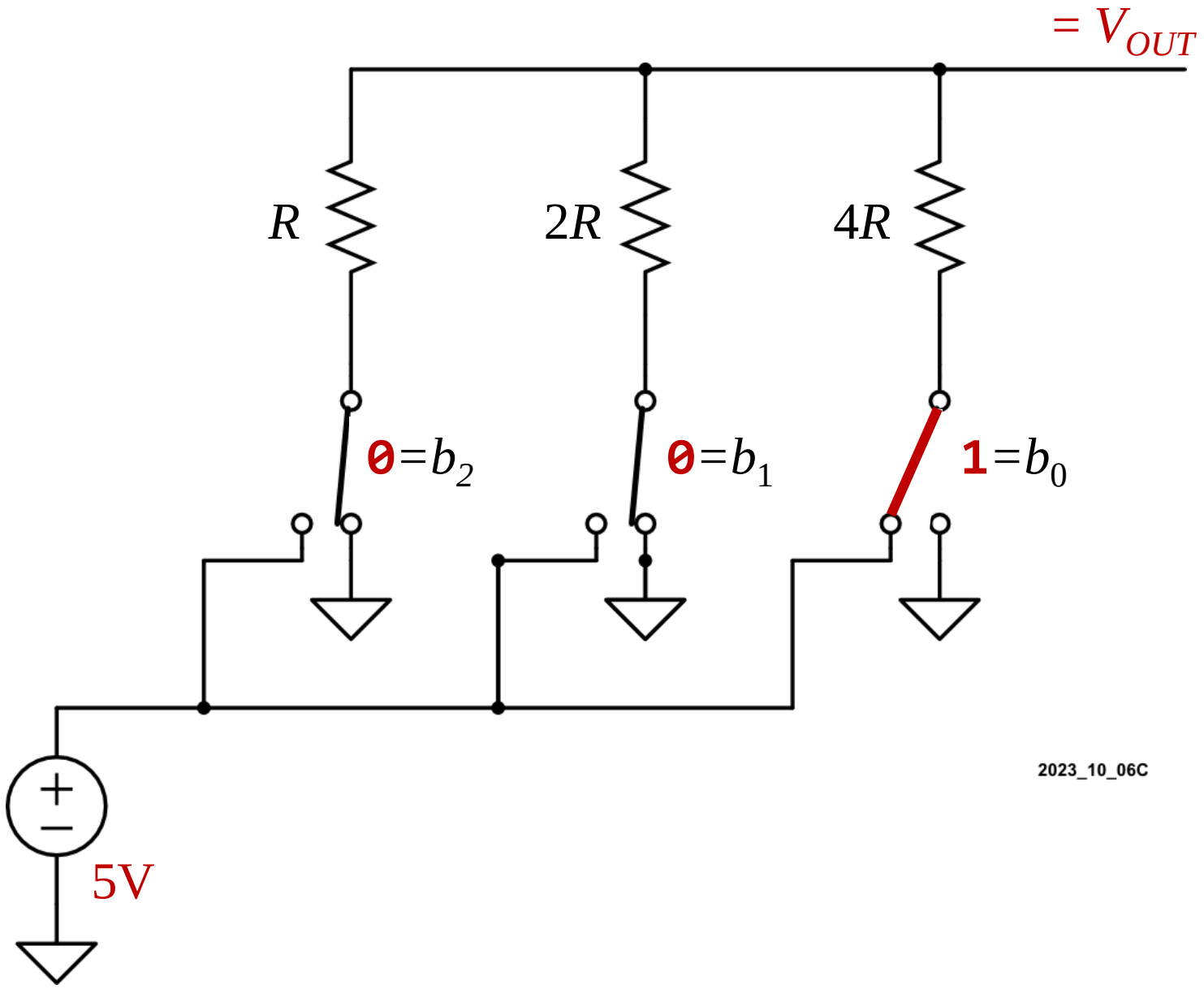
Binary-Scaled DAC

- Similar to a R-2R DAC, but resistors are weighted



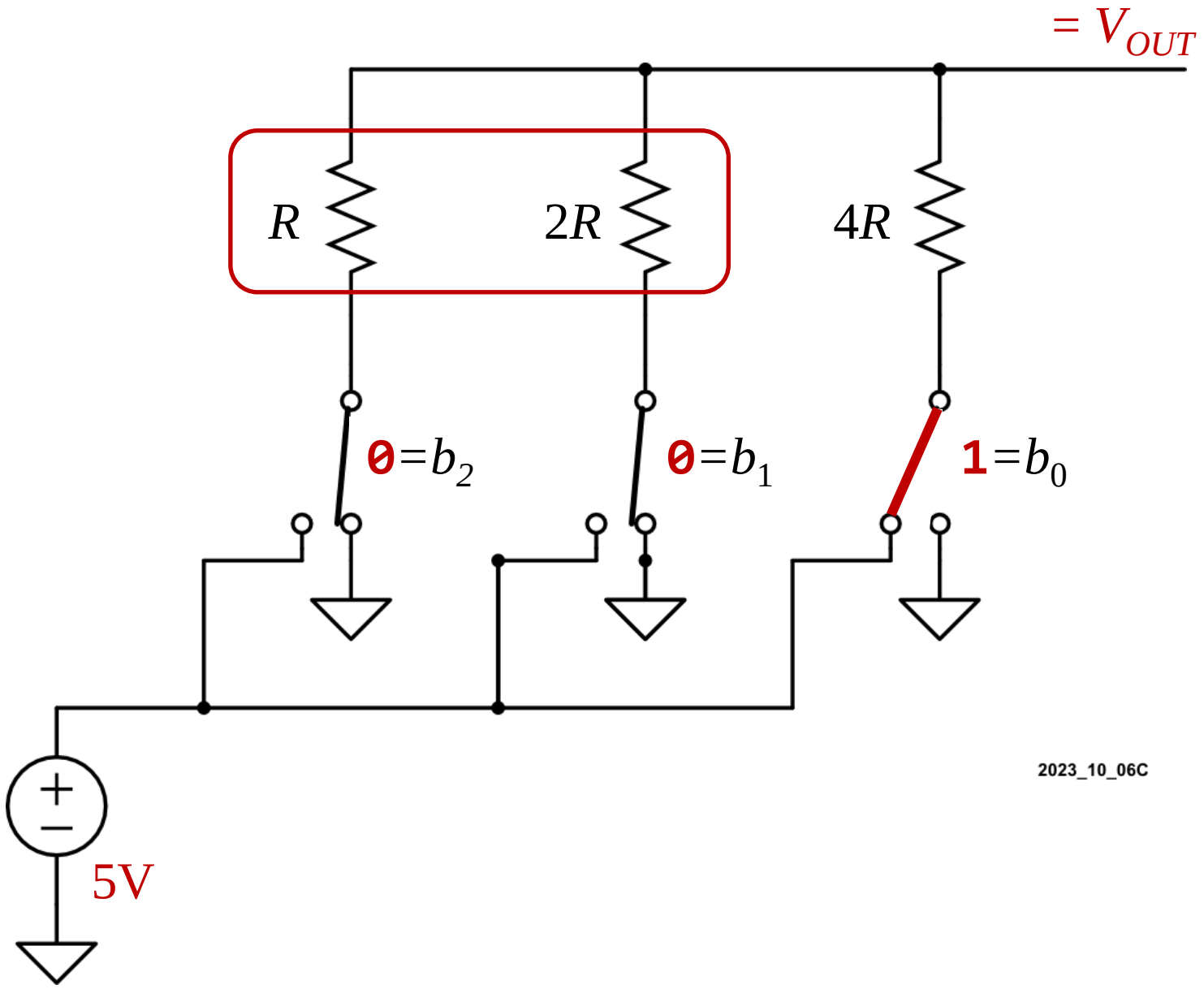
Binary-Scaled DAC

- Similar to a R-2R DAC, but resistors are weighted



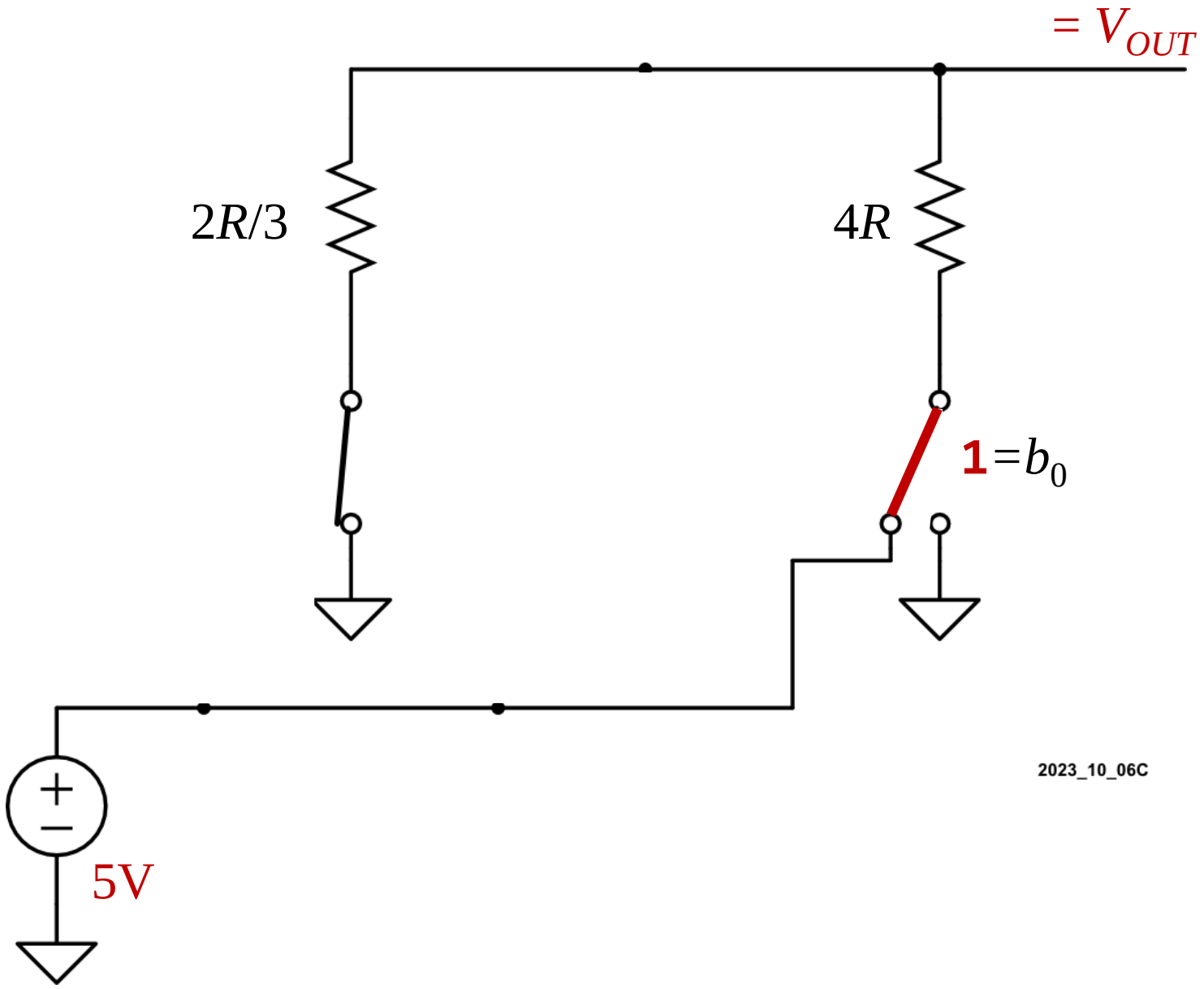
Binary-Scaled DAC

- Similar to a R-2R DAC, but resistors are weighted



Binary-Scaled DAC

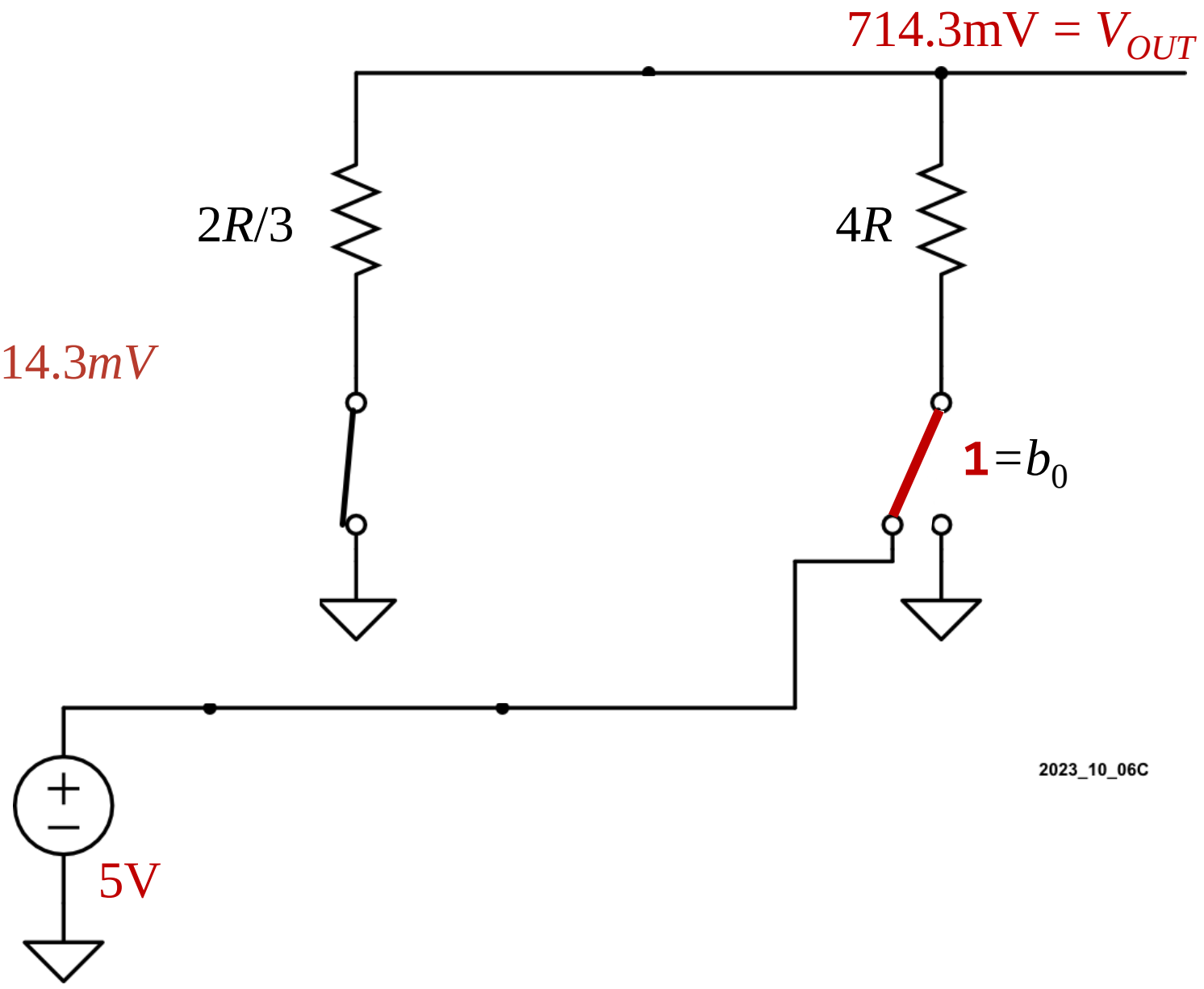
- Similar to a R-2R DAC, but resistors are weighted



Binary-Scaled DAC

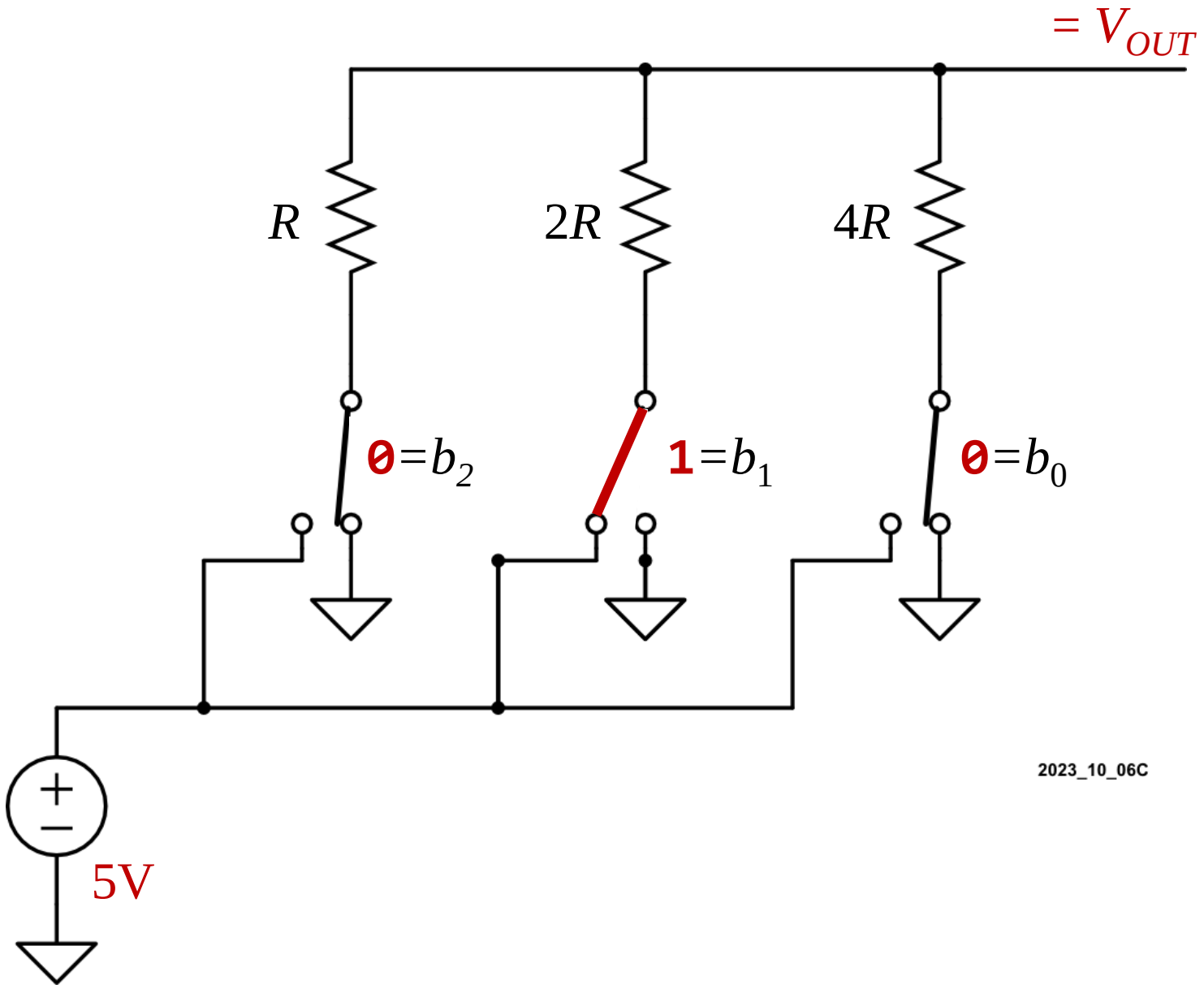
- Similar to a R-2R DAC, but resistors are weighted

$$V_{OUT} = 5V\left(\frac{2R/3}{4R + 2R/3}\right) = 5V\left(\frac{1}{7}\right) = 714.3mV$$



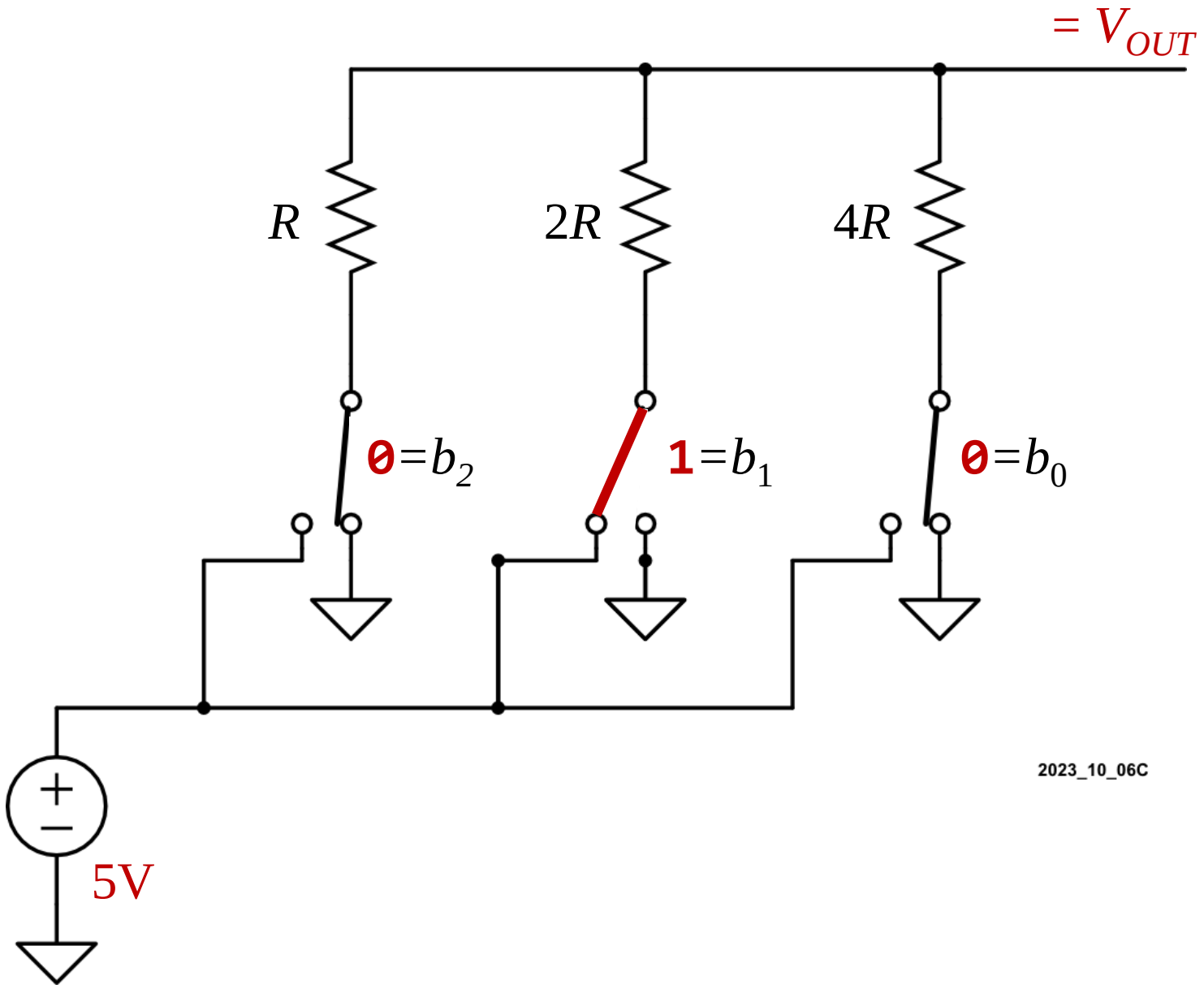
Binary-Scaled DAC

- Similar to a R-2R DAC, but resistors are weighted



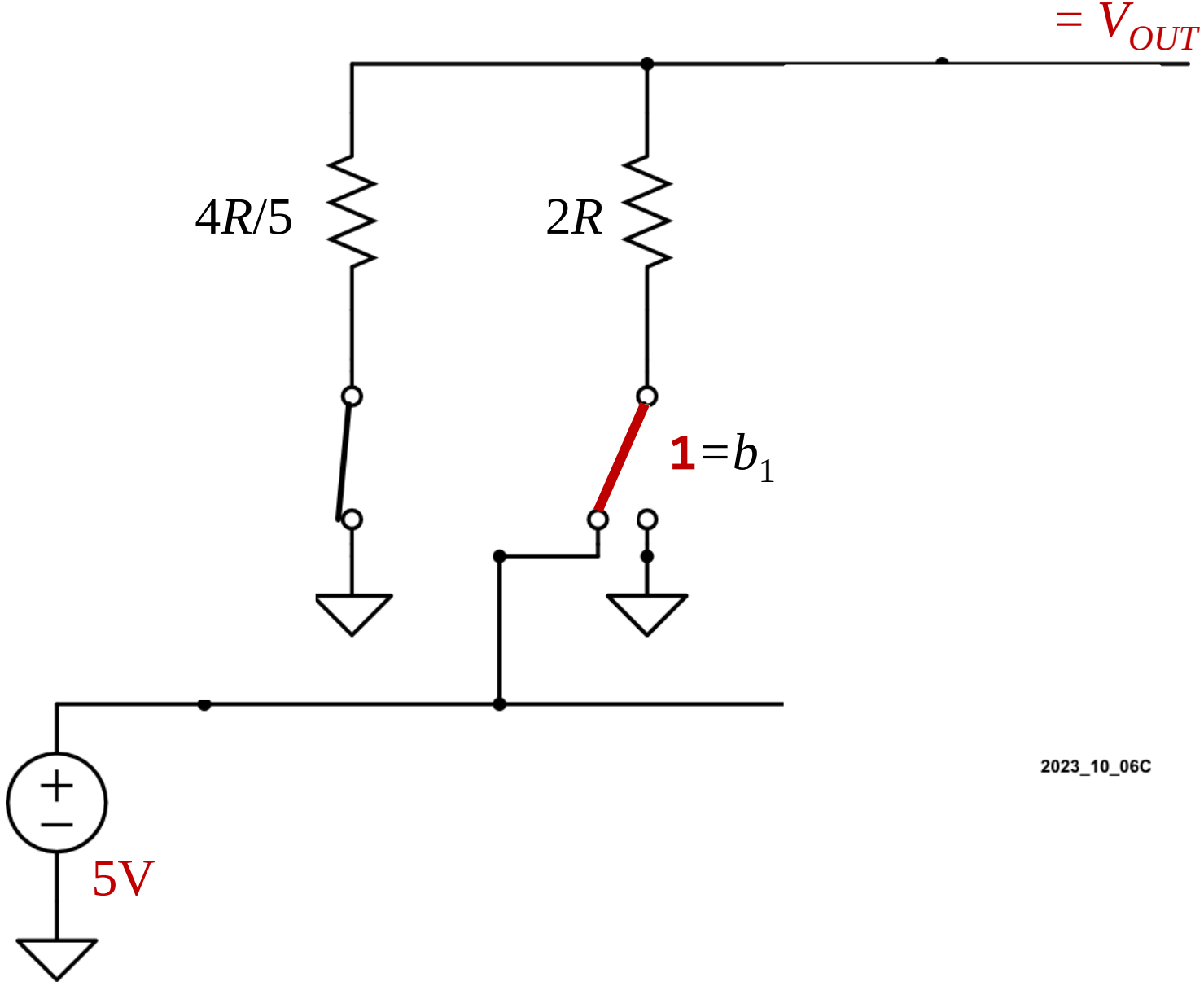
Binary-Scaled DAC

- Similar to a R-2R DAC, but resistors are weighted



Binary-Scaled DAC

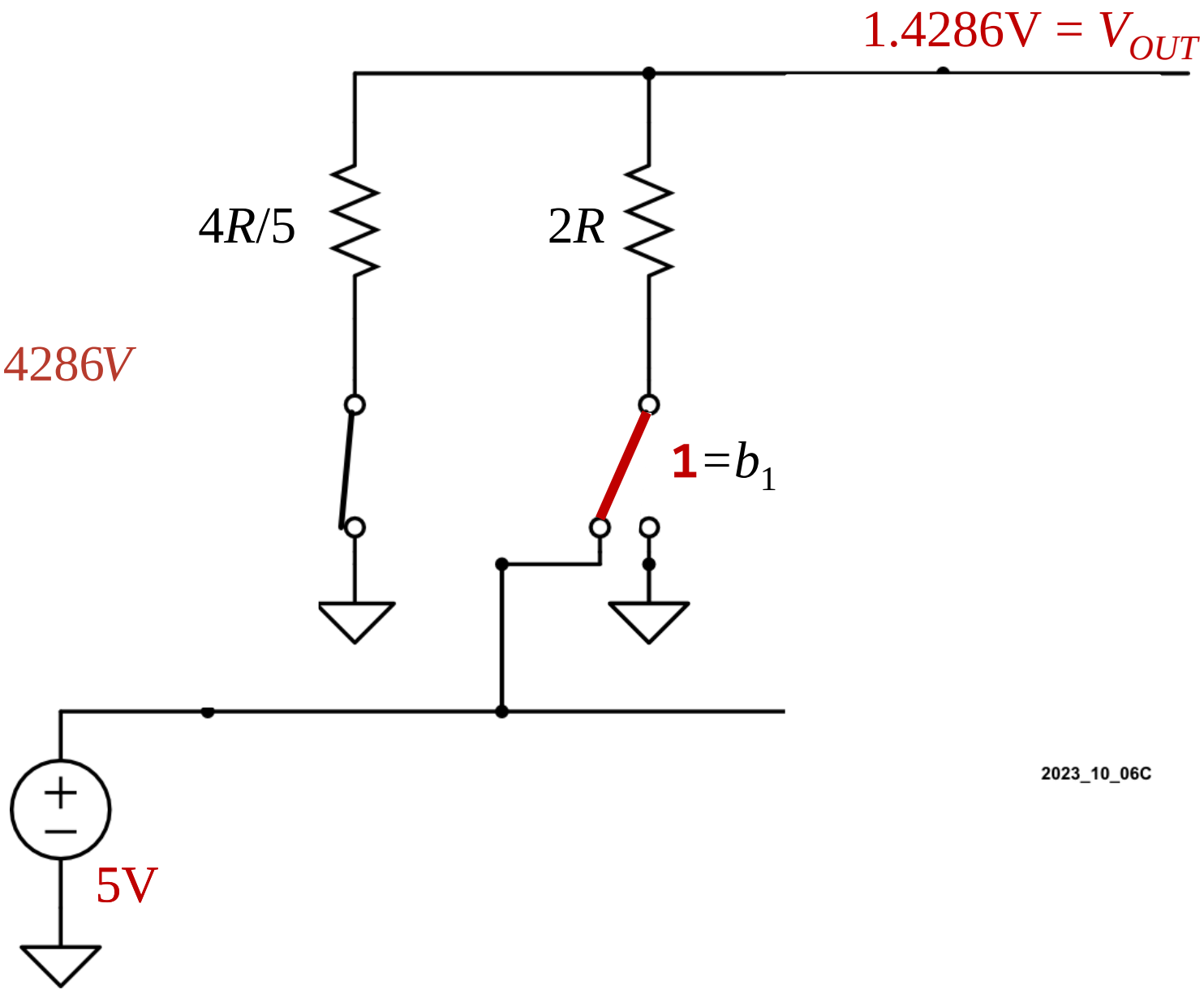
- Similar to a R-2R DAC, but resistors are weighted



Binary-Scaled DAC

- Similar to a R-2R DAC, but resistors are weighted

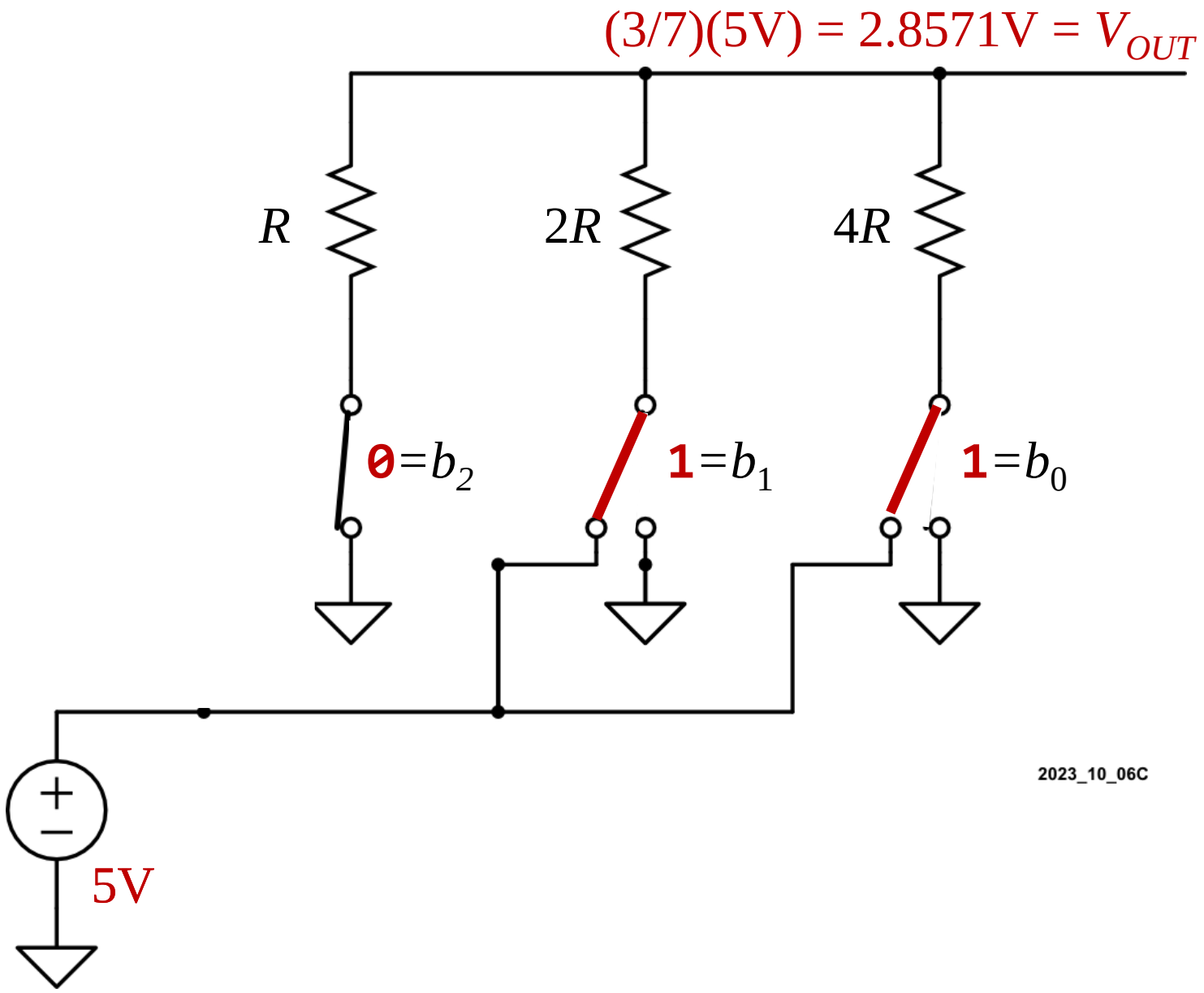
$$V_{OUT} = 5V\left(\frac{4R/5}{2R + 4R/5}\right) = 5V\left(\frac{2}{7}\right) = 1.4286V$$



2023_10_06C

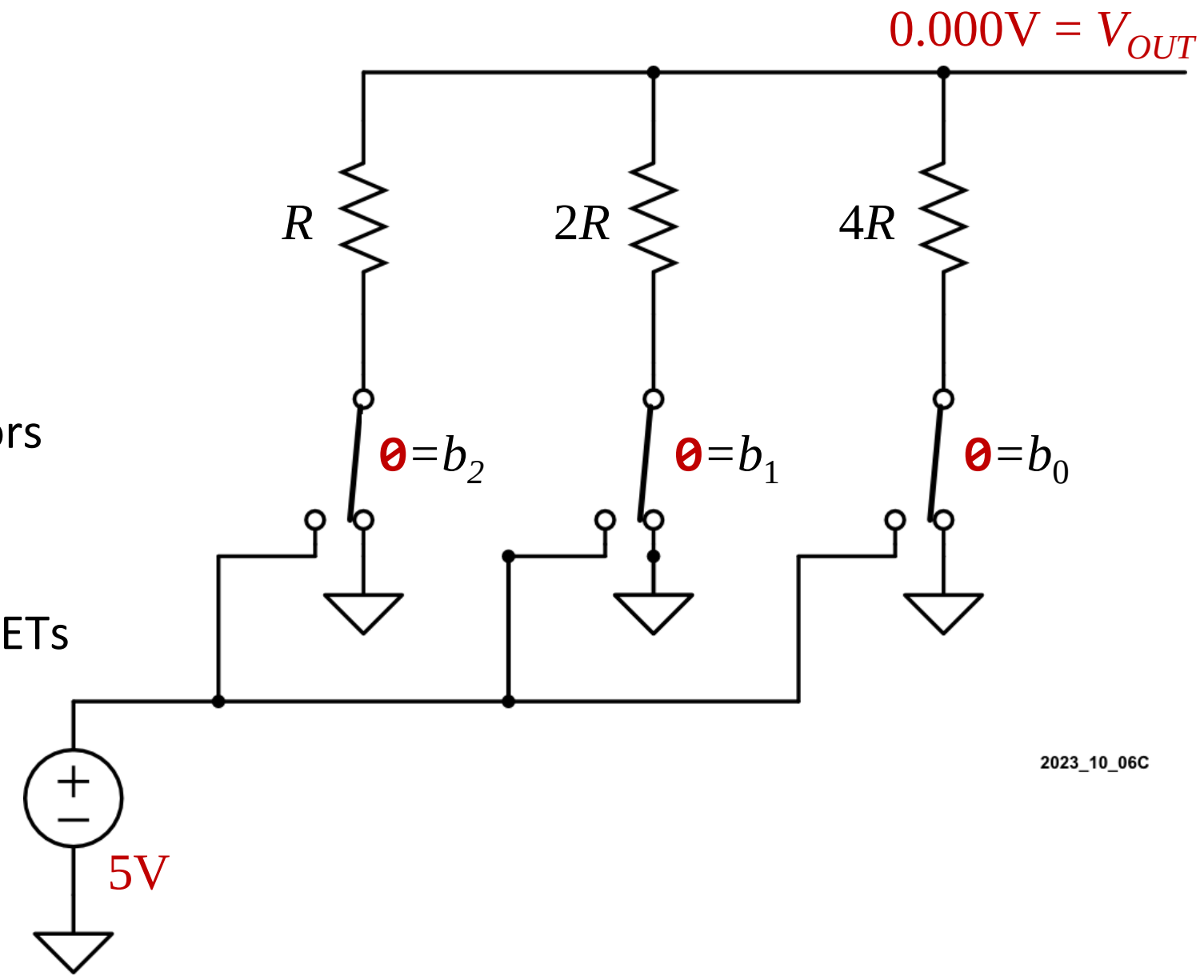
Binary-Scaled DAC

- Similar to a R-2R DAC, but resistors are weighted

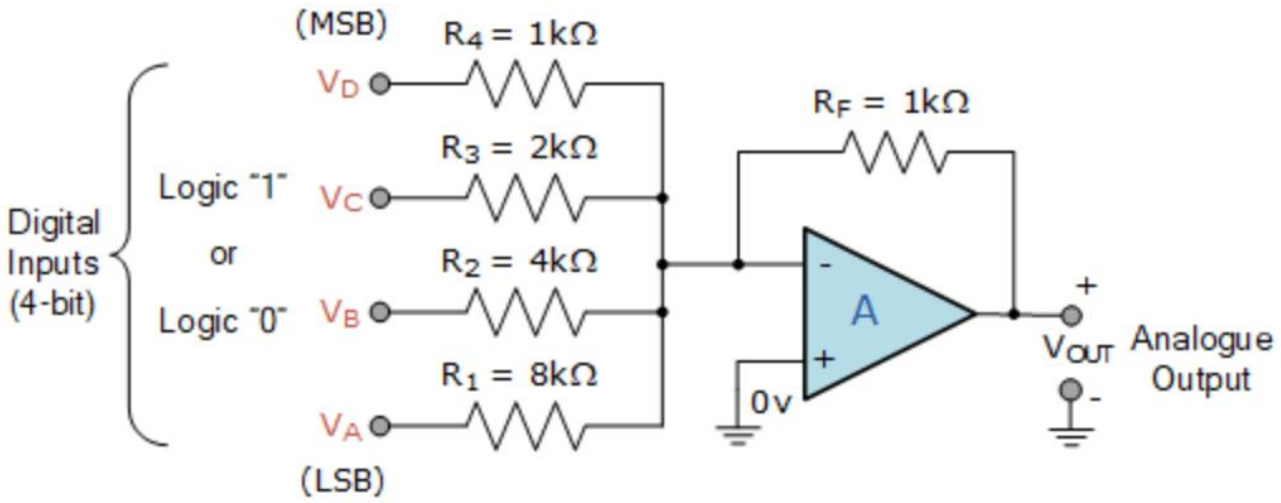


Binary-Scaled DAC

- Advantages:
  - It is simple in construction
  - It provides fast conversion
- Disadvantages:
  - Requires large range of resistors
  - Requires high precision for low value resistors.
  - Requires low resistance MOSFETs
  - Can be expensive
  - Use an op amp buffer to increase current drive



Binary-Scaled DAC



$$V_{OUT} = -\left[\frac{R_F}{R_4}V_D + \frac{R_F}{R_3}V_C + \frac{R_F}{R_2}V_B + \frac{R_F}{R_1}V_A\right]$$

$$V_{OUT} = -\left[\frac{1k\Omega}{1k\Omega}V_D + \frac{1k\Omega}{2k\Omega}V_C + \frac{1k\Omega}{4k\Omega}V_B + \frac{1k\Omega}{8k\Omega}V_A\right]$$

$$V_{OUT} = -\left[1V_D + \frac{1}{2}V_C + \frac{1}{4}V_B + \frac{1}{8}V_A\right]$$

https://www.electronics-tutorials.ws/combination/digital-to-analogue-converter.html

Remember, for All DACs

- Almost always will need an op amp buffer to increase current drive ($I_{OUT,MAX} < 1\text{mA}$)

Remember, for All DACs

- Almost always will need an op amp buffer to increase current drive ($I_{OUT,MAX} < 1\text{mA}$)
- Almost always will also need a low pass filter on the output to avoid glitches

Remember, for All DACs

- Almost always will need an op amp buffer to increase current drive ($I_{OUT,MAX} < 1\text{mA}$)

- Almost always will also need a low pass filter on the output to avoid glitches
- For example, 5V, 3-bit ADC
  - **011** input will be approximately (3/8)(5V) = 1.875V
  - We want to update the output to **100** $\rightarrow$ (4/8)(5V) = 2.500V

Remember, for All DACs

- Almost always will need an op amp buffer to increase current drive ($I_{OUT,MAX} < 1\text{mA}$)

- Almost always will also need a low pass filter on the output to avoid glitches
- For example, 5V, 3-bit ADC
  - **`011`** input will be approximately (3/8)(5V) = 1.875V
  - We want to update the output to **`100`** $\rightarrow$ (4/8)(5V) = 2.500V
  - What happens if most significant input bit switches first?

Remember, for All DACs

- Almost always will need an op amp buffer to increase current drive ($I_{OUT,MAX} < 1\text{mA}$)

- Almost always will also need a low pass filter on the output to avoid glitches
- For example, 5V, 3-bit ADC
  - **011** input will be approximately (3/8)(5V) = 1.875V
  - We want to update the output to **100** → (4/8)(5V) = 2.500V
  - What happens if most significant input bit switches first?

    011 → (3/8)(5V) = 1.875V

    111 → (7/8)(5V) = 4.375V

    110 → (6/8)(5V) = 3.750V

    100 → (4/8)(5V) = 2.500V