# Embedded Systems

## Digital-to-Analog & Analog-to-Digital Converters

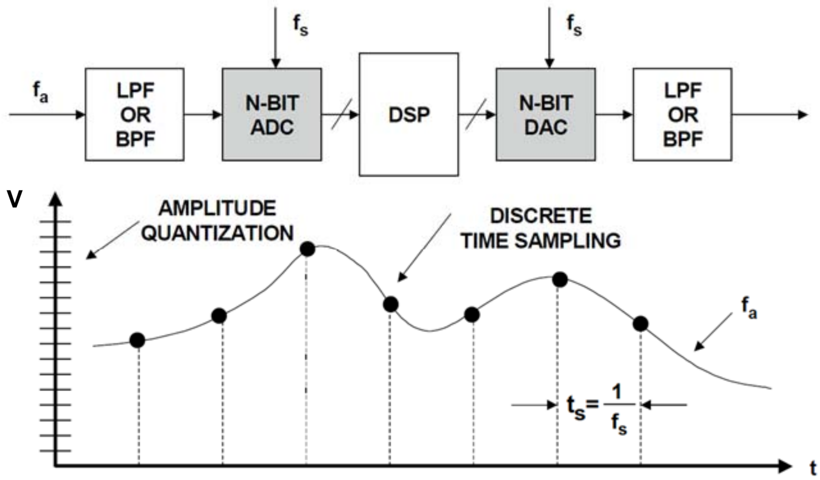


## A/D and D/A Converters

- Two basic types of converters:
  - Digital-to-analog (DACs or D/As)
  - Analog-to-digital (ADCs or A/Ds)
- Frequently utilized in embedded systems
  - Often integrated into µC (ATmega88: 10-bit ADC, but no DAC)
- Applications include:
  - Digital signal processing (DSP): audio, radio, …
  - Imaging (CCD)
  - Medical applications
  - Instrumentation, data loggers, …
  - Digital offset and gain adjustment
  - Sensors: temperature, weight bridge, …
  - Control systems: motor, temperature, digital servos, …
  - Communications, GPS, …
  - …

## Application Example - Digital Signal Processing

- E.g., digital audio filter, software-defined radio (SDR), …

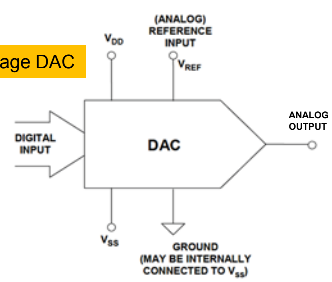


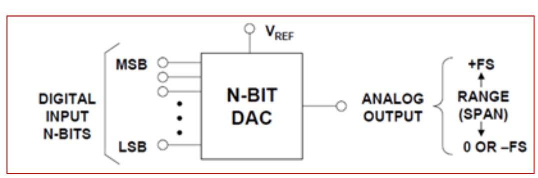



## Converters

- **DACs** convert a digital input value to a voltage or current
  - Voltage/current is a proportion of a reference voltage/current
  - The proportion is defined by the digital input applied

Voltage DAC

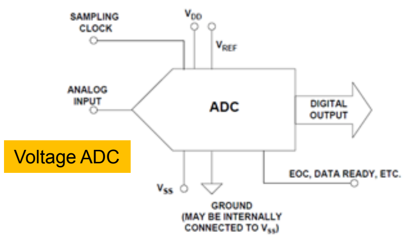


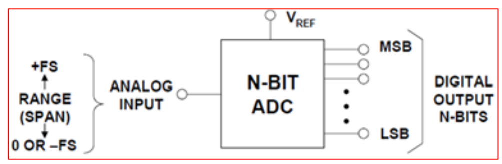




FS … Full-Scale

## Converters

- **ADCs convert a voltage or current to a digital output value**
  - **A digital representation of the analog voltage/current that is applied to the ADC input is outputted**
  - **The representation is proportional to a reference voltage/current**






## Pre- and Post-Processing & Accuracy vs Resolution

- **ADC → voltages and currents must be "normalized" to ranges compatible with ADC input ranges**

  **→often a preprocessing of input signals is required**

- **DAC → analog output voltages or currents from DACs are direct and in normalized form**

  **→ signals often need post-processing (e.g., scaled/amplified, filtered, etc.).**

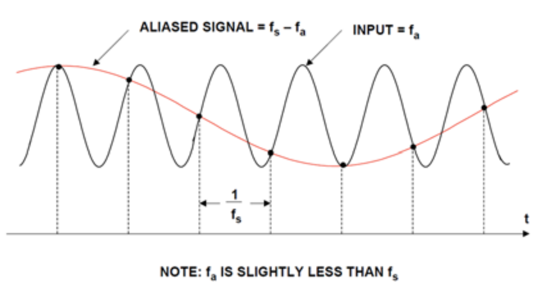
## A/D and D/A Converters

- **Different architectures**
- **It is important to select a suitable A/D or D/A converter for a given application → design requirements**
  - Processing speed
  - Resolution
  - Accuracy, linearity, …
  - Power consumption
  - …
- ***ADC architectures:***
  - successive approximation, flash, dual slope, sigma-delta, …
- ***DAC architectures:***
  - string, R-2R, PWM, …



## Aliasing and Sampling Frequency

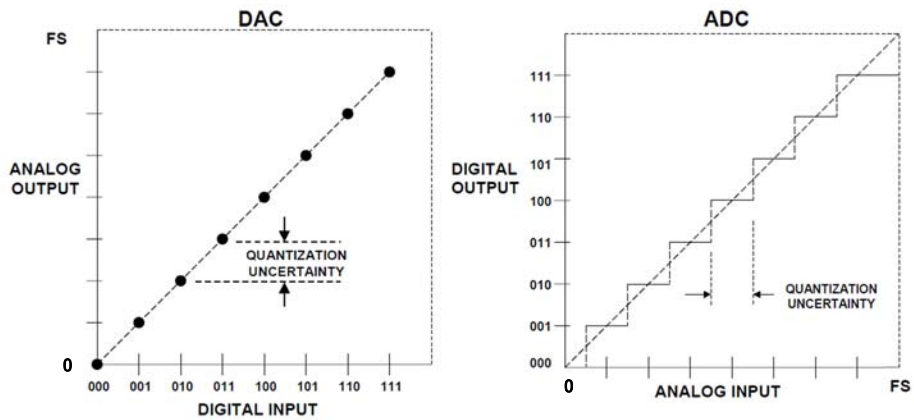
- **The required conversion speed depends on the application (e.g., sampling/reproducing an audio signal, thermocouples, HF transceiver, …)**
- **A signal with a maximum BANDWIDTH $f_a$ must be sampled at a rate $f_s > 2 f_a$ , or information about the signal will be lost because of aliasing.**




→ Oscilloscope

## Resolution of Data Converters

- **Transfer Functions for Ideal 3-Bit DAC and ADC**




→ Resolution affects quantization uncertainty → signal quantization errors



## Resolution of Data Converters

- **The digital input or output can have 1 to 32 bits**
  - **The greater the number of bits, the finer the resolution**
- **Example:**
  - **The 10-bit ADC used by the Atmega88 with a 5 V reference (FS)**
    - Has $2^{10} = 1024$ possible output codes
    - Can resolve $5/2^{10} = 5/1024 = 4.88$ mV

**Typically:**

$$\frac{Code_{Out/In}}{2^N} = \frac{S_{In/Out}}{FS}$$

fractional binary

**$Code_{Out/In}$ … code produced by ADC or code sent to DAC**
**N … number of bits**
**$S_{in/Out}$ … input signal to ADC or output signal of DAC**
**FS … full-scale (often equivalent to reference)**

## Resolution of Data Converters

- The resolution of data converters may be expressed in several different ways
- Quantization: The Size of a Least Significant Bit (LSB)

| RESOLUTION N | $2^N$ | VOLTAGE (10V FS) | ppm FS | % FS | dB FS |
|---|---|---|---|---|---|
| 2-bit | 4 | 2.5 V | 250,000 | 25 | – 12 |
| 4-bit | 16 | 625 mV | 62,500 | 6.25 | – 24 |
| 6-bit | 64 | 156 mV | 15,625 | 1.56 | – 36 |
| 8-bit | 256 | 39.1 mV | 3,906 | 0.39 | – 48 |
| 10-bit | 1,024 | 9.77 mV (10 mV) | 977 | 0.098 | – 60 |
| 12-bit | 4,096 | 2.44 mV | 244 | 0.024 | – 72 |
| 14-bit | 16,384 | 610 μV | 61 | 0.0061 | – 84 |
| 16-bit | 65,536 | 153 μV | 15 | 0.0015 | – 96 |
| 18-bit | 262,144 | 38 μV | 4 | 0.0004 | – 108 |
| 20-bit | 1,048,576 | 9.54 μV (10 μV) | 1 | 0.0001 | – 120 |
| 22-bit | 4,194,304 | 2.38 μV | 0.24 | 0.000024 | – 132 |
| 24-bit | 16,777,216 | 596 nV* | 0.06 | 0.000006 | – 144 |

*600nV is the Johnson Noise in a 10kHz BW of a 2.2kΩ Resistor @ 25°C



## Example - D/A Converter

- An engineer needs to select a DAC that can produce an output voltage in the range of 0 to 3 V. The voltage increment ΔV needs to be <= 1 mV.
- Q1: How many bits (N) will be needed to meet this specification?
- Q2: What will be the value of ΔV ?

## Fractional Binary Numbers

- For ADCs & DACs **fractional binary coding** is utilized, which is always normalized to full-scale (→ ref. voltage)
- Integer binary can be interpreted as fractional binary if all integer values are divided by 2^N

**WHOLE NUMBERS:**

$$\text{Number}_{10} = \underset{\text{MSB}}{a_{N-1}}2^{N-1} + a_{N-2}2^{N-2} + \ldots + a_1 2^1 + \boxed{\underset{\text{LSB}}{a_0}2^0}$$

Example: $1011_2 = (1\times2^3) + (0\times2^2) + (1\times2^1) + (1\times2^0)$
$= 8 + 0 + 2 + 1 = 11_{10}$

**FRACTIONAL NUMBERS:**

$$\text{Number}_{10} = \underset{\text{MSB}}{a_{N-1}}2^{-1} + a_{N-2}2^{-2} + \ldots + a_1 2^{-(N-1)} + \underset{\text{LSB}}{a_0}2^{-N}$$

Example: $0.1011_2 = (1\times0.5) + (0\times0.25) + (1\times0.125) + (1\times0.0625)$
$= 0.5 + 0 + 0.125 + 0.0625 = 0.6875_{10}$

N=4

Embedded Systems, ECE:3360. The University of Iowa, 2019

Slide 13

## Unipolar Codes

- Coding method is related to the
  - analog input range of an ADC or
  - analog output range of a DAC
- The simplest case is when the input to the ADC or the output of the DAC is always a unipolar (positive) voltage/current

Example:
4-bit converter

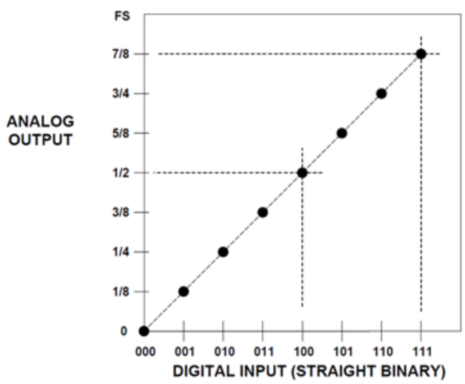
| BASE 10 NUMBER | SCALE | +10 V FS | BINARY |
|---|---|---|---|
| +15 | +FS – 1 LSB = 15/16 FS | 9.375 | 1111 |
| +14 | +7/8 FS | 8.750 | 1110 |
| +13 | +13/16 FS | 8.125 | 1101 |
| +12 | +3/4 FS | 7.500 | 1100 |
| +11 | +11/16 FS | 6.875 | 1011 |
| +10 | +5/16 FS | 6.250 | 1010 |
| +9 | +9/16 FS | 5.625 | 1001 |
| +8 | +1/2 FS | 5.000 | 1000 |
| +7 | +7/16 FS | 4.375 | 0111 |
| +6 | +3/8 FS | 3.750 | 0110 |
| +5 | +5/16 FS | 3.125 | 0101 |
| +4 | +1/4 FS | 2.500 | 0100 |
| +3 | +3/16 FS | 1.875 | 0011 |
| +2 | +1/8 FS | 1.250 | 0010 |
| +1 | 1 LSB = +1/16 FS | 0.625 | 0001 |
| 0 | 0 | 0.000 | 0000 |

Embedded Systems, ECE:3360. The University of Iowa, 2019

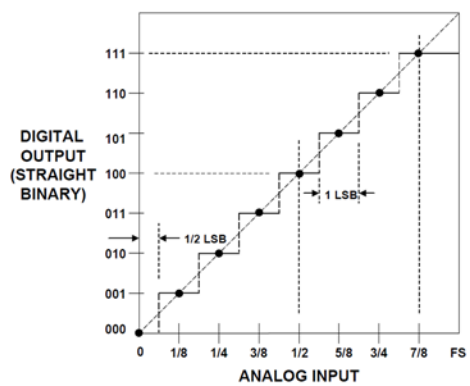
Slide 14

# Unipolar Codes

DAC

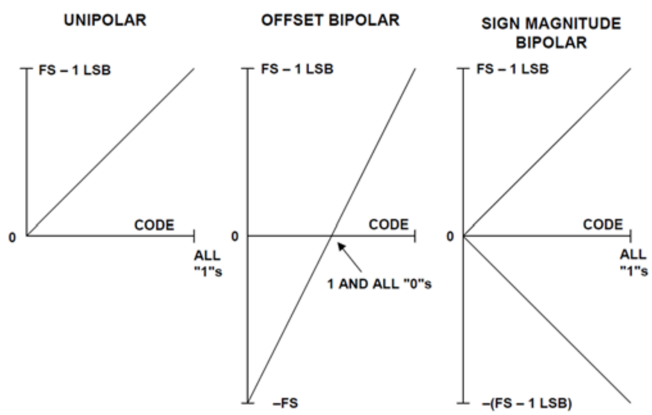



ADC






# Unipolar and Bipolar DACs



## Bipolar Codes

- **For some applications it is desirable to represent both positive and negative analog quantities with binary codes**
- Either *offset binary*, *two's complement*, *ones complement*, and *sign magnitude* codes will accomplish this, but <u>*offset binary* and *two's complement* are by far the most popular</u>

Example:
4-bit converter

| BASE 10 NUMBER | SCALE | ±5V FS | OFFSET BINARY | TWOS COMP. | ONES COMP. | SIGN MAG. |
|---|---|---|---|---|---|---|
| +7 | +FS – 1LSB = +7/8 FS | +4.375 | 1111 | 0111 | 0111 | 0111 |
| +6 | +3/4 FS | +3.750 | 1110 | 0110 | 0110 | 0110 |
| +5 | +5/8 FS | +3.125 | 1101 | 0101 | 0101 | 0101 |
| +4 | +1/2 FS | +2.500 | 1100 | 0100 | 0100 | 0100 |
| +3 | +3/8 FS | +1.875 | 1011 | 0011 | 0011 | 0011 |
| +2 | +1/4 FS | +1.250 | 1010 | 0010 | 0010 | 0010 |
| +1 | +1/8 FS | +0.625 | 1001 | 0001 | 0001 | 0001 |
| 0 | 0 | 0.000 | 1000 | 0000 | *0000 | *1000 |
| –1 | – 1/8 FS | –0.625 | 0111 | 1111 | 1110 | 1001 |
| –2 | – 1/4 FS | –1.250 | 0110 | 1110 | 1101 | 1010 |
| –3 | – 3/8 FS | –1.875 | 0101 | 1101 | 1100 | 1011 |
| –4 | –1/2 FS | –2.500 | 0100 | 1100 | 1011 | 1100 |
| –5 | –5/8 FS | –3.125 | 0011 | 1011 | 1010 | 1101 |
| –6 | –3/4 FS | –3.750 | 0010 | 1010 | 1001 | 1110 |
| –7 | – FS + 1LSB = –7/8 FS | –4.375 | 0001 | 1001 | 1000 | 1111 |
| –8 | – FS | –5.000 | 0000 | 1000 | | |

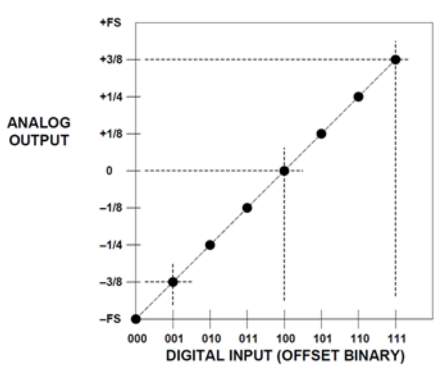
CODES NOT NORMALLY USED IN COMPUTATIONS

| * | ONES COMP. | SIGN MAG. |
|---|---|---|
| 0+ | 0000 | 0000 |
| 0– | 1111 | 1000 |

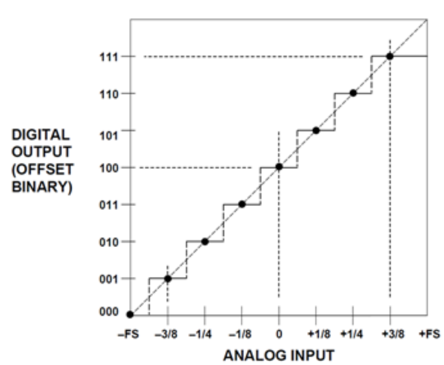


## Bipolar Codes

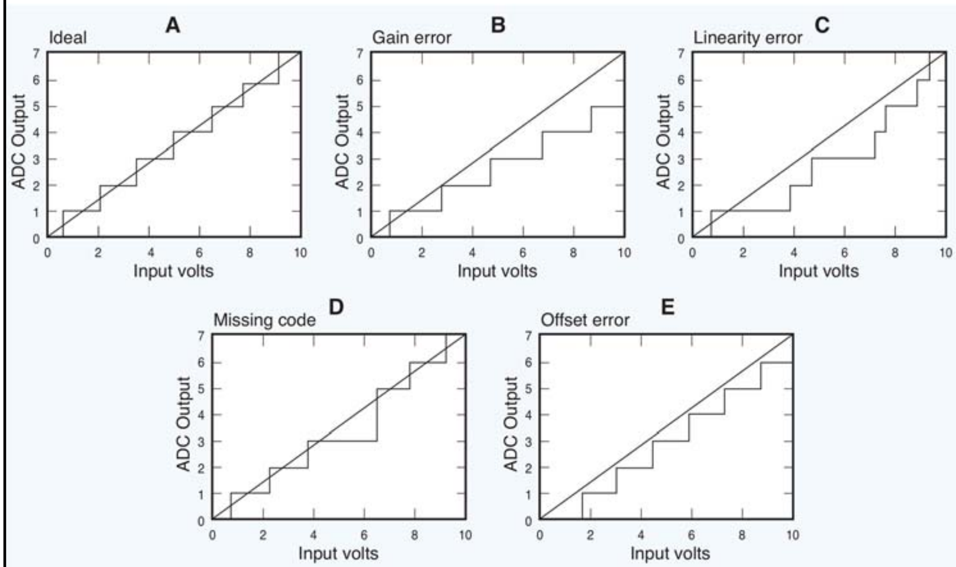
DAC




ADC



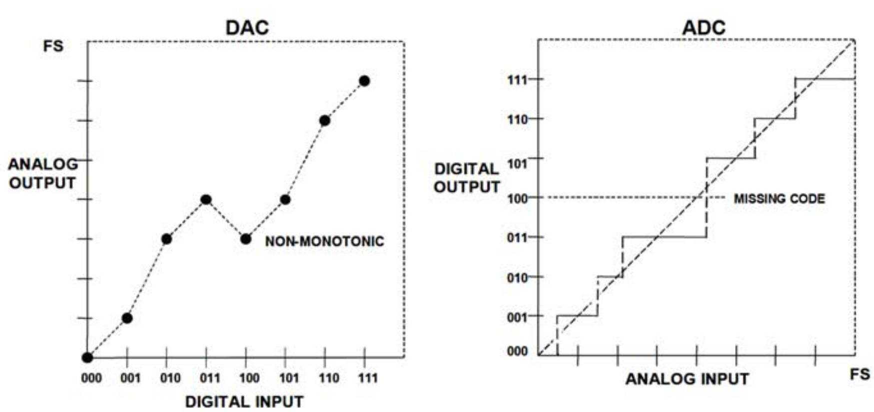
## ADC - Errors






## Non-Ideal Converters

- **Transfer Functions for Non-Ideal 3-Bit DAC and ADC**




*Transfer Functions for Non-Ideal 3-Bit DAC and ADC*

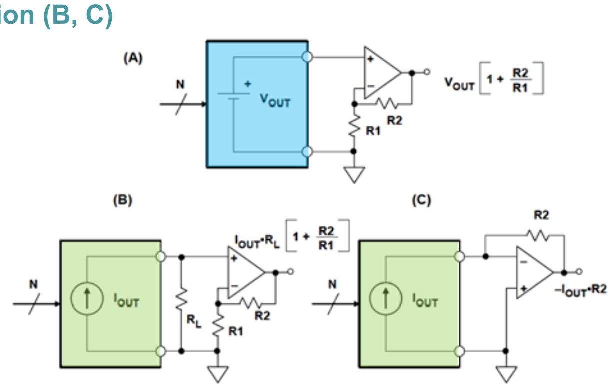
# DACs



# DACs

- **Many DACs require a voltage reference**
  - Effects DAC performance (e.g., accuracy, noise, drift, …)
- **Voltage DACs may require a buffer/amplifier (A)**
- **Some DACs output a current**

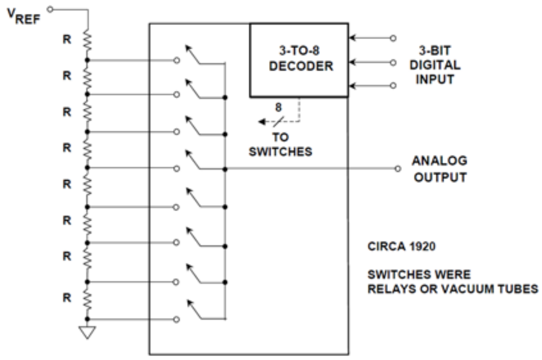
  → an op-amp can be used to perform the current-to-voltage (I/V) conversion (B, C)




*Buffering DAC Outputs with Op Amps*

## Simple Voltage-Output DAC

- A N-bit String DAC uses 2^N resistors and 2^N (CMOS) switches → high complexity for large N
- Nonlinear versions are also possible
- Output impedance depends on input → use a voltage buffer

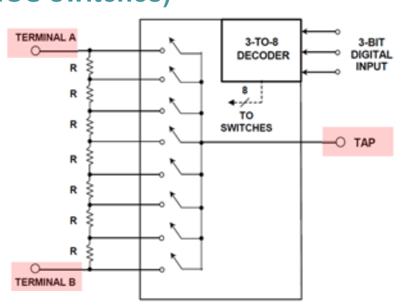



*Simplest Voltage-Output Thermometer DAC:*
*The Kelvin Divider*



## Digital Potentiometers

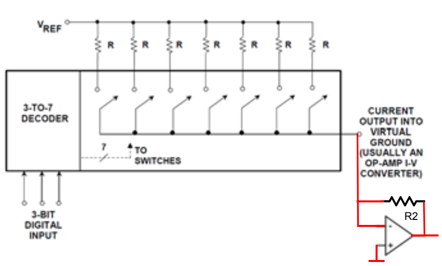
- Another variation of the string DAC is the digital potentiometer
  - Better "adjustability" than their mechanical counterparts
  - Immune to mechanical vibration and oxidation of the wiper contact
  - Adjustments can be made without human intervention
  - Typically, the voltage on the input pins cannot exceed the supplies (CMOS switches)

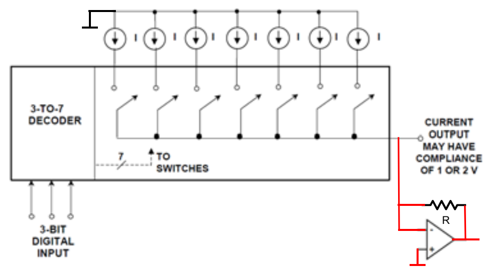



*A Slight Modification to a Kelvin DAC Yields a "Digital Potentiometer"*

## Thermometer Current-Output DAC

- **Analogous to a string DAC**
- **Consists of 2^N – 1 switchable current sources (resistors and a voltage reference or active current sources) connected to an output terminal**
- **Output must be at or close to ground → I-V converter**



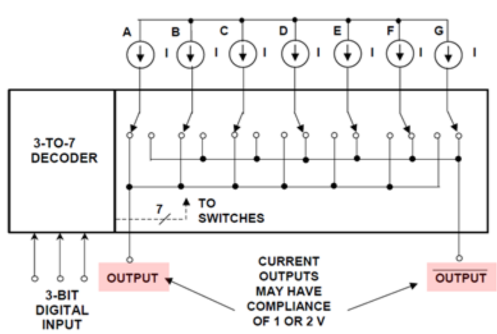

→ switching a current on and off is disruptive → glitches

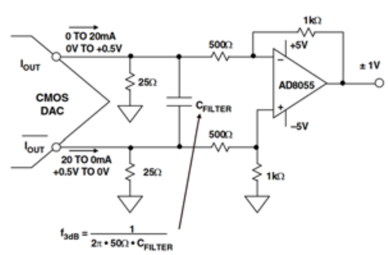
**Solution: complementary current outputs**



## Thermometer DAC with Complementary Current Outputs

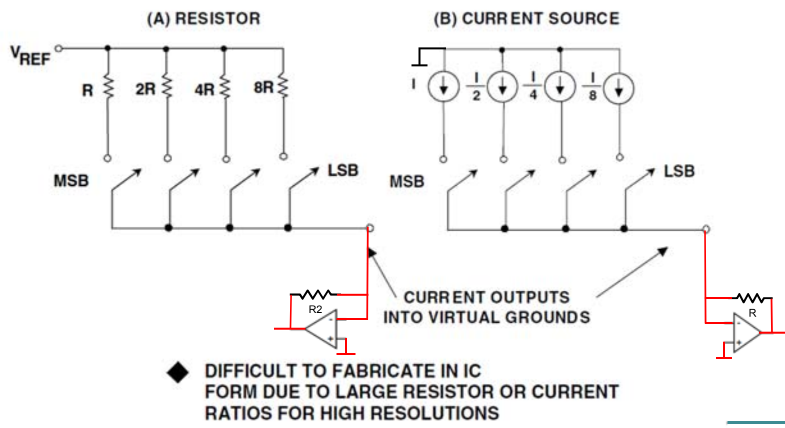
- **Two options for I-V conversion:**
  - Connect one line to ground → standard I-V converter
  - Differential DC-coupled I-V converter



## Current-Mode Binary-Weighted DACs

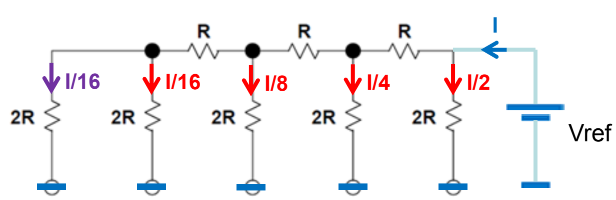
- **Implementation:**
  - N weighted resistors and a voltage reference or
  - N weighted current sources with
  - ratio 1:2:4: … :2^(N-1)





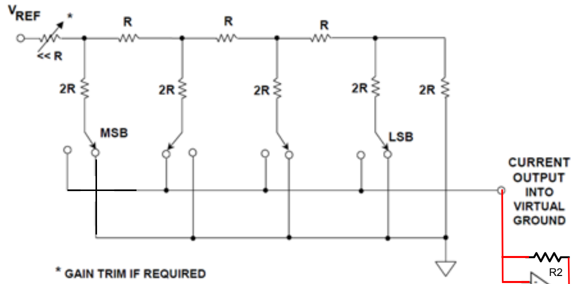

## R-2R Resistor Ladder

- **R-2R resistor ladder → the most common DAC building-block**
- **It uses resistors of only two different values (ratio is 1:2)**
- **An N-bit DAC → 2N resistors, which are quite easy to trim**



## R-2R Resistor Ladder based DAC

- **R-2R resistor ladder →**
  - Current output




  - Voltage output

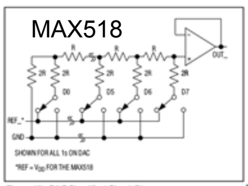
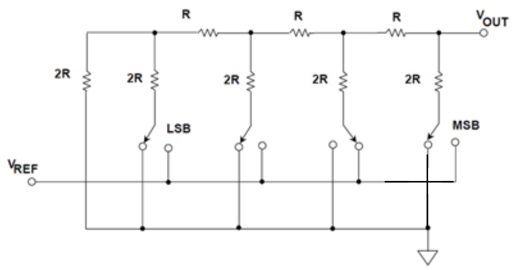
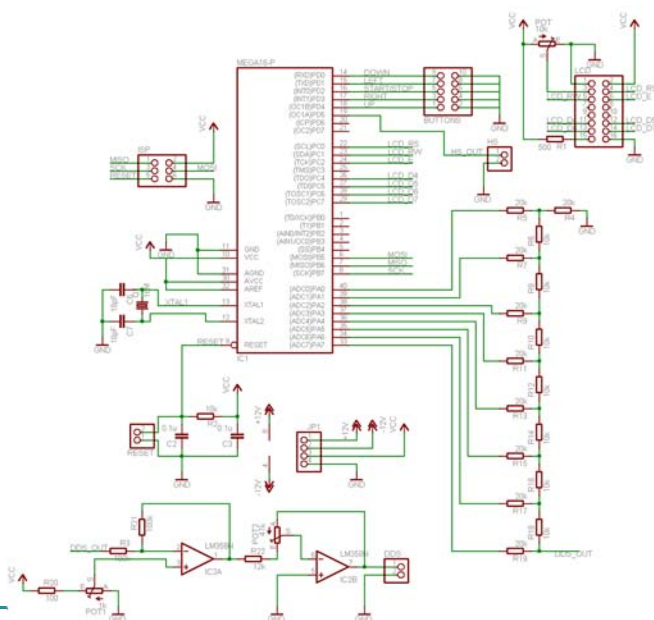



Figure 15. DAC Simplified Circuit Diagram



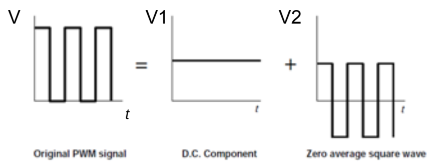



## Example - Simple R-2R DAC (DDS)

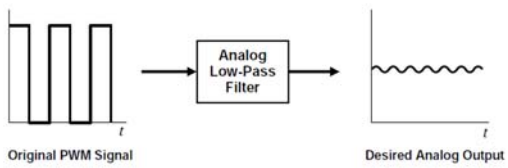


## DAC with Pulse Width Modulation (PWM)

- **PWM DAC**
  - basic idea → decomposition of a PWM signal

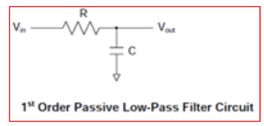


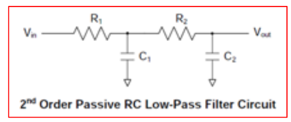

- **Analog Filtering of PWM Signal**




**Change of duty cycle → change of DC component**

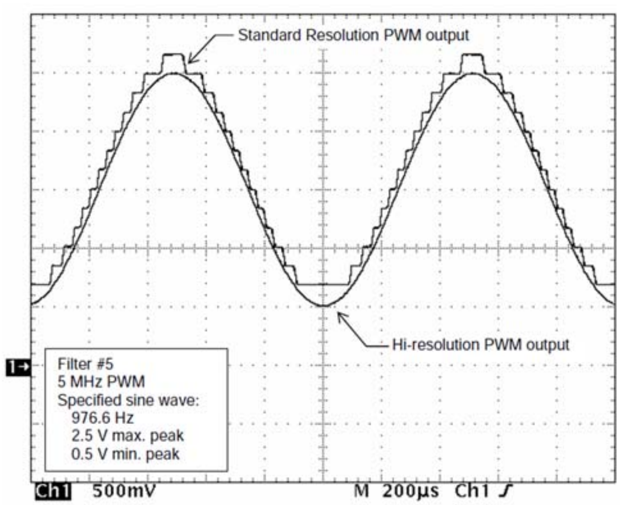
RC Low-Pass Filter






## Output of a PWM Sine Wave

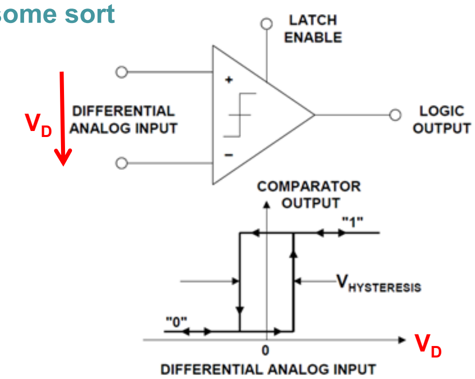


# ADC



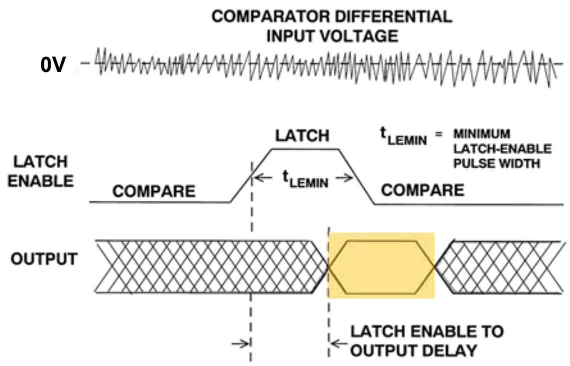
# The Comparator: A 1-Bit ADC

- The output will be a logic "1" if $V_D > 0$ and a logic "0" for $V_D < 0$
- Some devices have a hysteresis
- →useful building block for other ADC architectures
  - → there is no ADC architecture which does not use at least one comparator of some sort



## Comparator

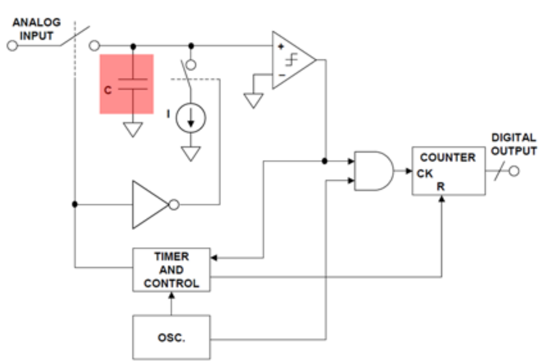
- Some comparators have an internal latch
- The latch-enable signal has two states:
  - compare (track)
  - latch (hold)





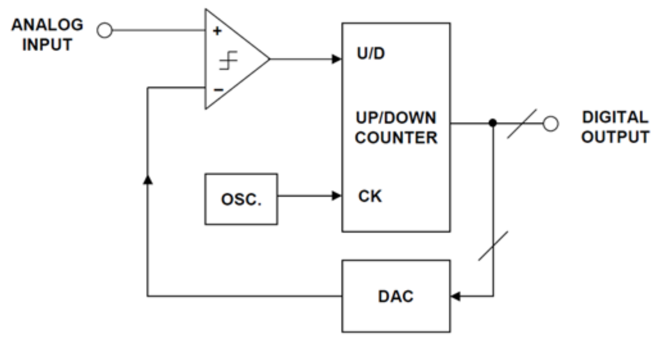

## Charge Run-Down ADCs

- First samples the analog input and stores the voltage on a fixed capacitor
- The capacitor is then discharged with a constant current source → the time required for complete discharge is measured using a counter
- Accuracy depends on: C, current source, time-base, …



## Tracking ADCs

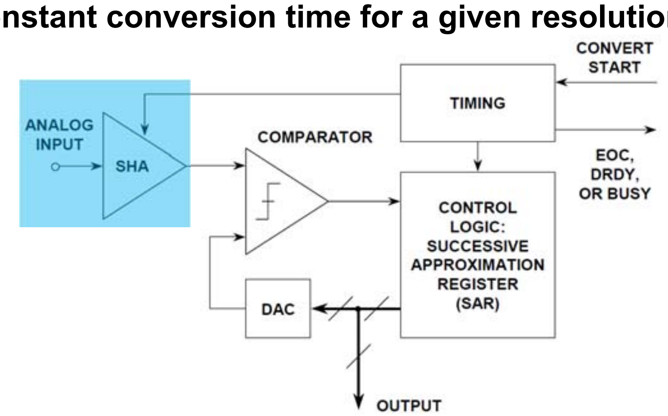
- Continually compares the input signal with a reconstructed representation of the input signal (DAC)
- The up/down counter is controlled by the comparator output.
- → slow step response → not widely used






## Successive Approximation ADCs

- Used in the ATmega88PA and other μCs
- The mainstay of data acquisition for many years
- Recent design improvements → sampling frequency in MHz-region
- → constant conversion time for a given resolution

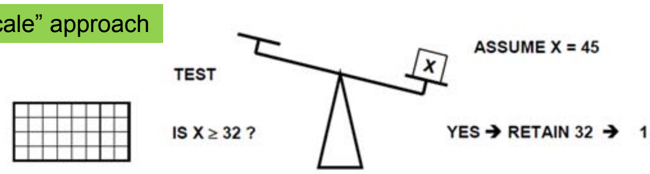


## Successive Approximation ADCs

- **Successive Approximation Algorithm (6-bit ADC)**

"Balance scale" approach

MSB=$2^5$

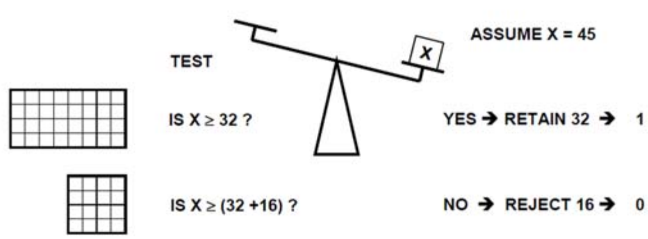





## Successive Approximation ADCs

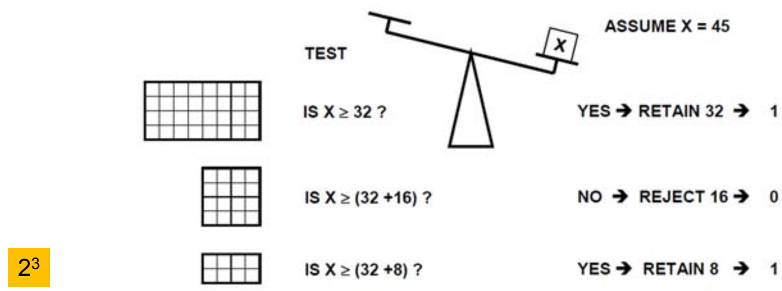
- **Successive Approximation Algorithm (6-bit ADC)**

$2^4$



## Successive Approximation ADCs

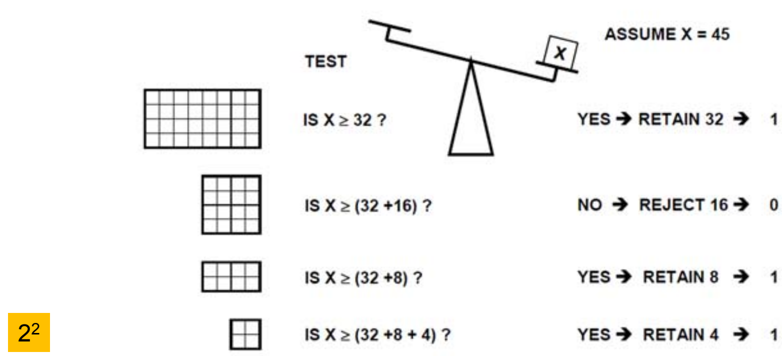
- **Successive Approximation Algorithm (6-bit ADC)**





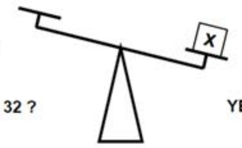

## Successive Approximation ADCs

- **Successive Approximation Algorithm (6-bit ADC)**



## Successive Approximation ADCs

- **Successive Approximation Algorithm (6-bit ADC)**



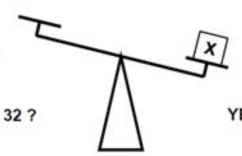

ASSUME X = 45

| | TEST | |
|---|---|---|
| | IS X ≥ 32 ? | YES → RETAIN 32 → 1 |
| | IS X ≥ (32 +16) ? | NO → REJECT 16 → 0 |
| | IS X ≥ (32 +8) ? | YES → RETAIN 8 → 1 |
| | IS X ≥ (32 +8 + 4) ? | YES → RETAIN 4 → 1 |
| $2^1$ | IS X ≥ (32 +8 + 4 + 2) ? | NO → REJECT 2 → 0 |




## Successive Approximation ADCs

- **Successive Approximation Algorithm (6-bit ADC)**

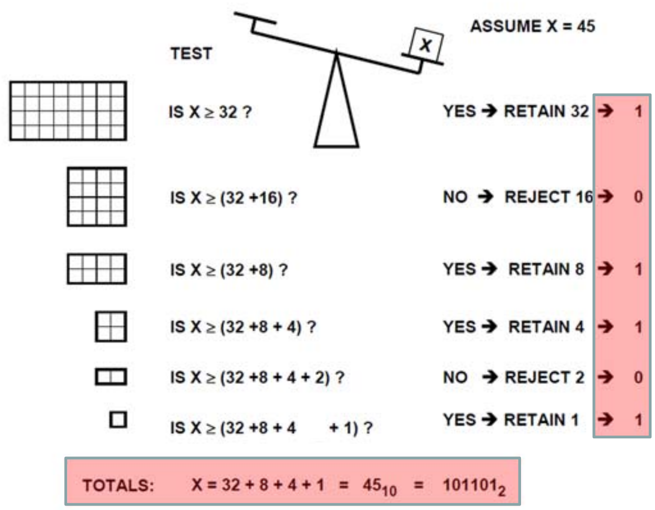

ASSUME X = 45

| | TEST | |
|---|---|---|
| | IS X ≥ 32 ? | YES → RETAIN 32 → 1 |
| | IS X ≥ (32 +16) ? | NO → REJECT 16 → 0 |
| | IS X ≥ (32 +8) ? | YES → RETAIN 8 → 1 |
| | IS X ≥ (32 +8 + 4) ? | YES → RETAIN 4 → 1 |
| | IS X ≥ (32 +8 + 4 + 2) ? | NO → REJECT 2 → 0 |
| LSB=$2^0$ | IS X ≥ (32 +8 + 4 + 1) ? | YES → RETAIN 1 → 1 |

TOTALS: X = 32 + 8 + 4 + 1 = $45_{10}$ = $101101_2$

## Successive Approximation ADCs

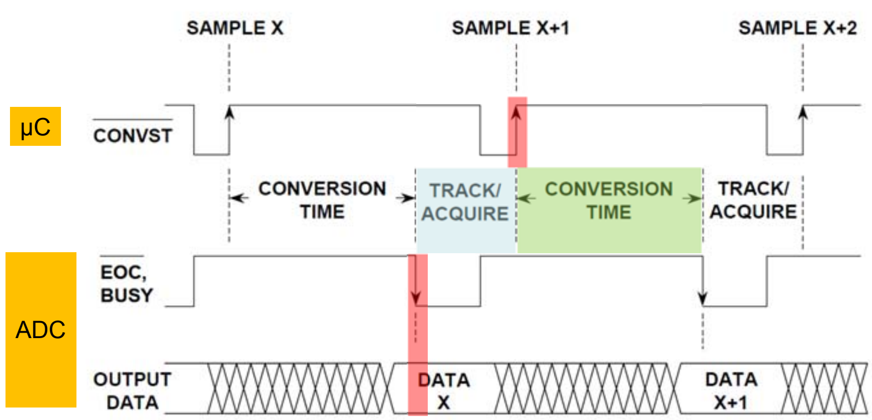
- **Successive Approximation Algorithm (6-bit ADC)**






## Successive Approximation ADCs

- **Input signal must be stable during conversion**
  - **Sample & hold stage**
- **Typical SAR ADC timing**



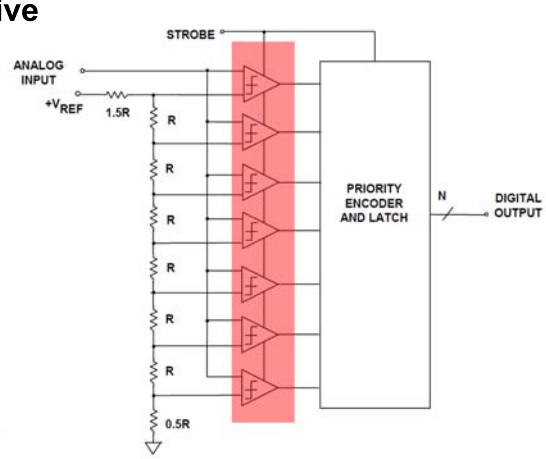
## Successive Approximation ADCs

- **Often integrated on µCs**
  - **ATmega88:**
    8/10-bit SAR ADC
    Up to 76.9 kSPS (up to 15 kSPS at max. resolution)
- **Advantages**
  - **High Accuracy**
  - **Typically Low Power**
  - **Low cost**
- **Disadvantages**
  - **Max sample rates approximately 2-5 MHz**
  - **Medium resolution**



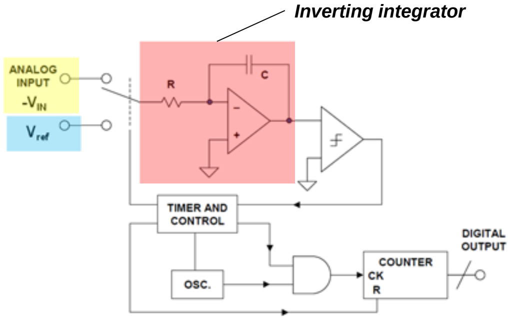
## Flash Converters

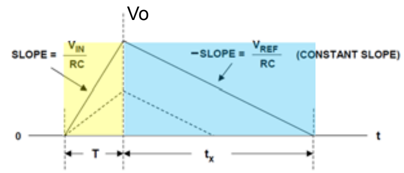
- **Flash (parallel) ADCs are the fastest type of ADC**
- **Have a large numbers of components:**
  - **N-bit flash ADC consists of 2^N resistors and 2^N–1 comparators**
- **Very fast, but lower resolution, high power dissipation, and expensive**



## Dual Slope ADCs

- "Breakthrough" in ADCs for high resolution applications such as digital voltmeters (DVMs), etc.
- Many advantages: conversion accuracy is independent of C and clock frequency, fixed input signal integration period → noise rejection (e.g., 50/60 Hz)
- Disadvantage: slow




$$\frac{V_{IN}}{RC}\,T = \frac{V_{REF}}{RC}\,t_x$$

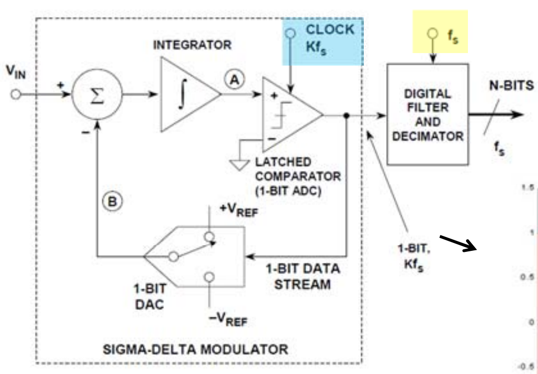
$$t_x = \frac{V_{IN}}{V_{REF}}\,T$$

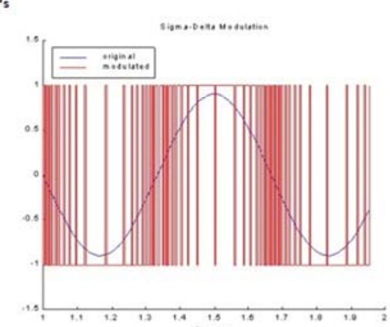
HIGH NORMAL MODE REJECTION AT MULTIPLES OF $\frac{1}{T}$



## Sigma-Delta (or Delta-Sigma) ADC

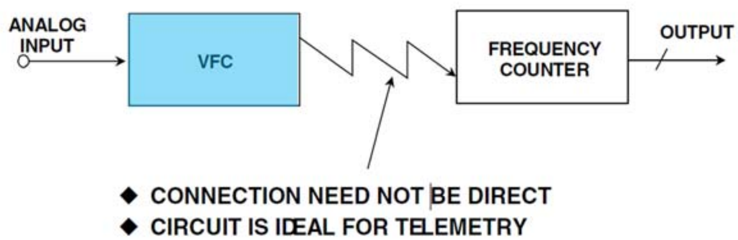
- Conversion approach: k-fold oversampling (1-bit signal) and digital filtering with down-conversion
- High resolution, high stability (averages and filters out noise), low power, moderate cost, and low to medium conversion speed



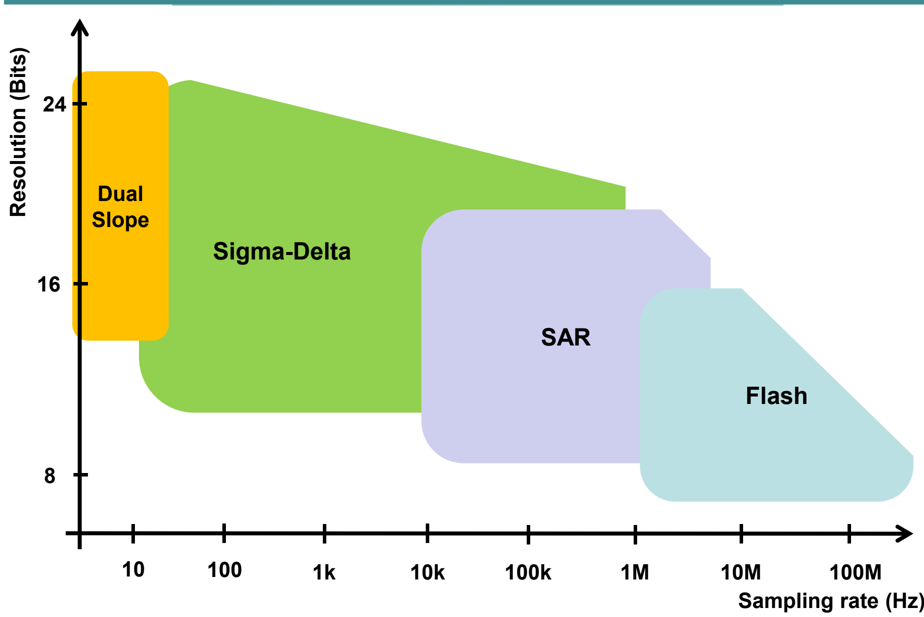
## Voltage-to-Frequency Converter

- A voltage-to-frequency converter (VFC) is an oscillator whose frequency is linearly proportional to a control voltage
- VFCs are typically small, cheap, and low-powered
- Can be used in combination with an optocoupler for isolation → e.g., industrial applications






## Comparison of ADC Architectures



… EOL