Analog to Digital Conversion

Why It's Needed

Embedded systems often need to measure values of physical parameters

These parameters are usually continuous (*analog*) and not in a digital form which computers (which operate on discrete data values) can process

- Temperature
  - Thermometer (do you have a fever?)
  - Thermostat for building, fridge, freezer
  - Car engine controller
  - Chemical reaction monitor
  - Safety (e.g. microprocessor processor thermal management)
- Light (or infrared or ultraviolet) intensity
  - Digital camera
  - IR remote control receiver
  - Tanning bed
  - UV monitor
- Rotary position
  - Wind gauge
  - Knobs
- Pressure
  - Blood pressure monitor
  - Altimeter
  - Car engine controller
  - Scuba dive computer
  - Tsunami detector
- Acceleration
  - Air bag controller
  - Vehicle stability
  - Video game remote
- Mechanical strain
- Other
  - Touch screen controller
  - EKG, EEG
  - Breathalyzer

The Big Picture

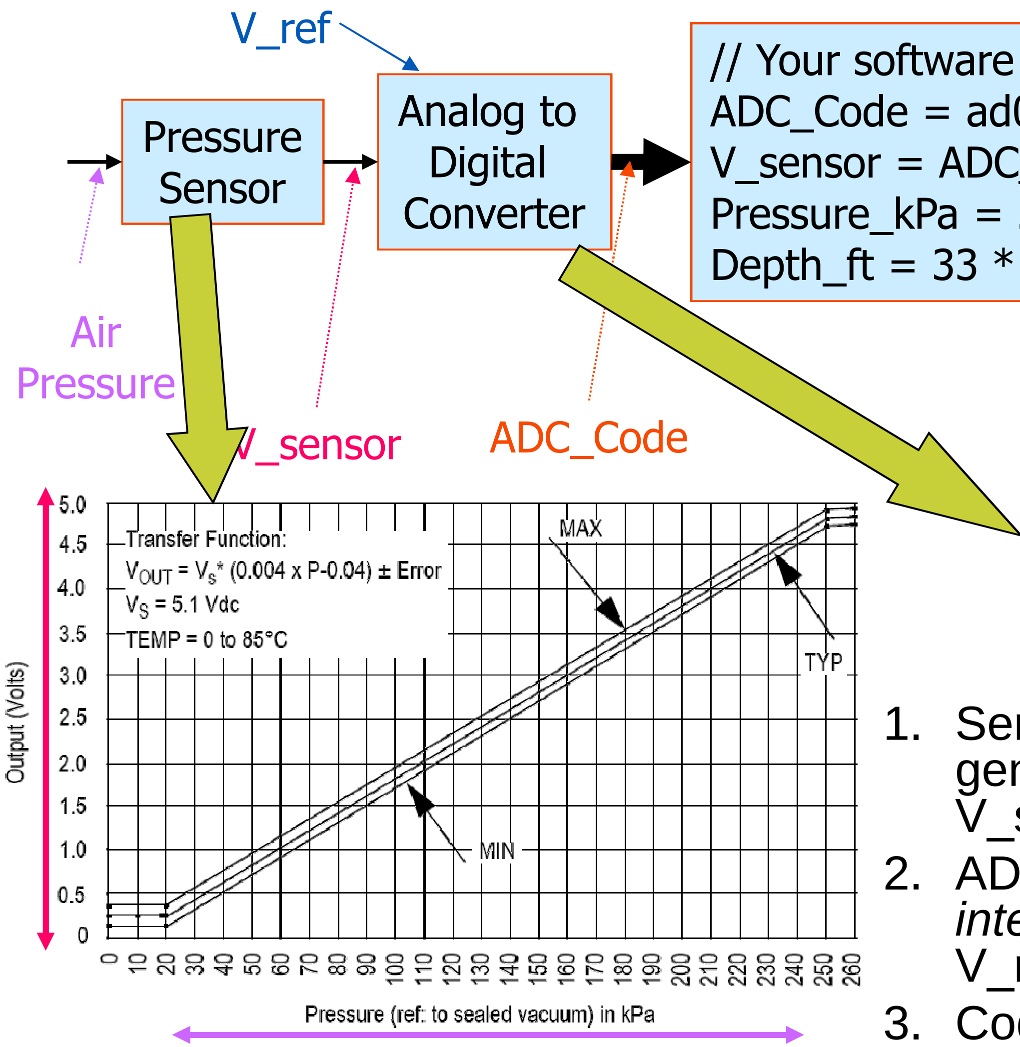
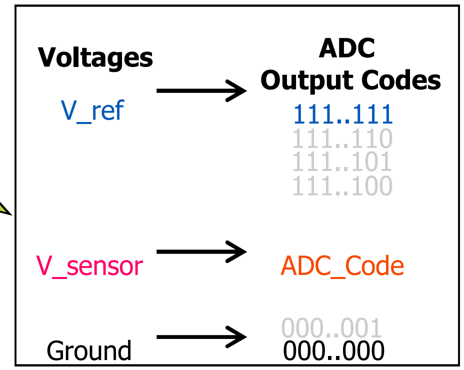


Figure 4. Output vs. Absolute Pressure

```
// Your software
ADC_Code = ad0;
V_sensor = ADC_code*V_ref/1023;
Pressure_kPa = 250 * (V_sensor/V_supply+0.04);
Depth_ft = 33 * (Pressure_kPa – Atmos_Press_kPa)/101.3;
```

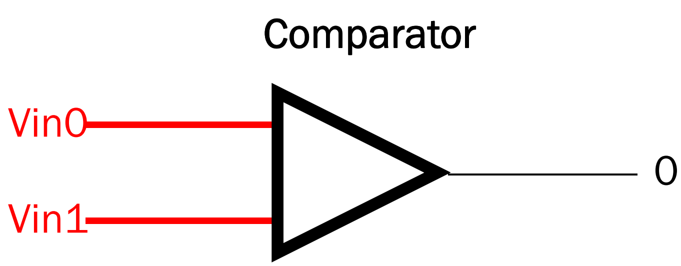


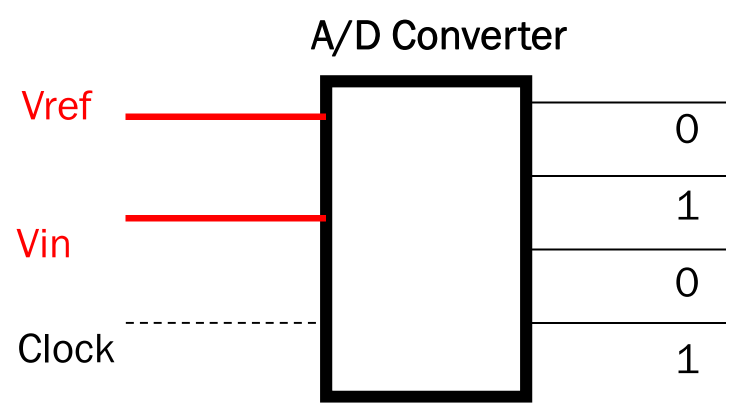
1. Sensor detects *air pressure* and generates a proportional *output voltage* V_sensor
2. ADC generates a proportional digital *integer* (code) based on V_sensor and V_ref
3. Code can convert that integer to a something more useful
   1. first a float representing the *voltage*,
   2. then another float representing *pressure*,
   3. finally another float representing *depth*

Getting From Analog to Digital

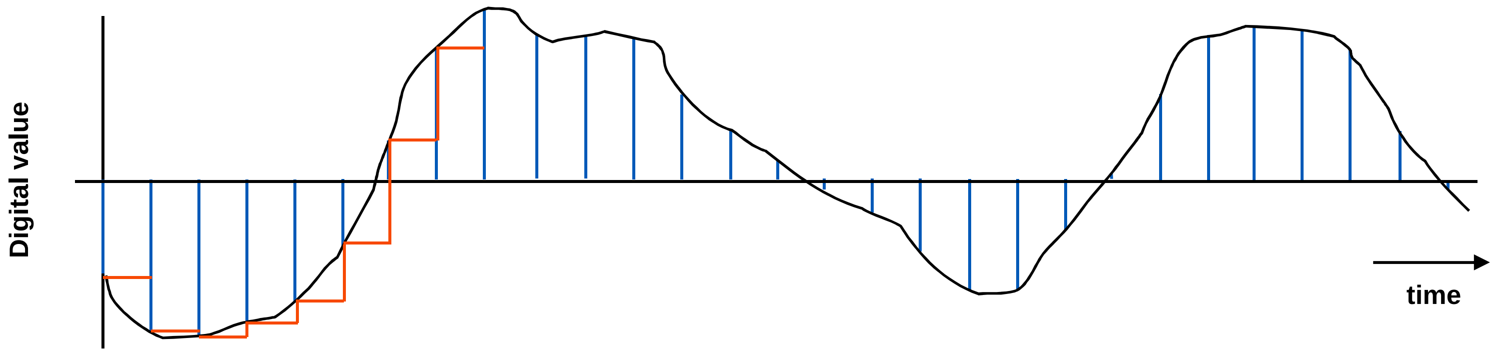
A **Comparator** is a circuit which compares an analog input voltage with a reference voltage and determines which is larger, returning a 1-bit number

An **Analog to Digital converter** [AD or ADC] is a circuit which accepts an analog input signal (usually a voltage) and produces a corresponding multi-bit number at the output.





Waveform Sampling and Quantization



A waveform is **sampled** at a constant rate – every Δ_t

- Each such sample represents the instantaneous amplitude at the instant of sampling
- *"At 37 ms, the input is 1.91341914513451451234311… V"*
- Sampling converts a **continuous time** signal to a **discrete time** signal

The sample can now be **quantized** (converted) into a digital value

- Quantization represents a **continuous** (analog) value with the closest **discrete** (digital) value
- *"The sampled input voltage of 1.91341914513451451234311… V is best represented by the code 0x018, since it is in the range of 1.901 to 1.9980 V which corresponds to code 0x018."*

Transfer Function

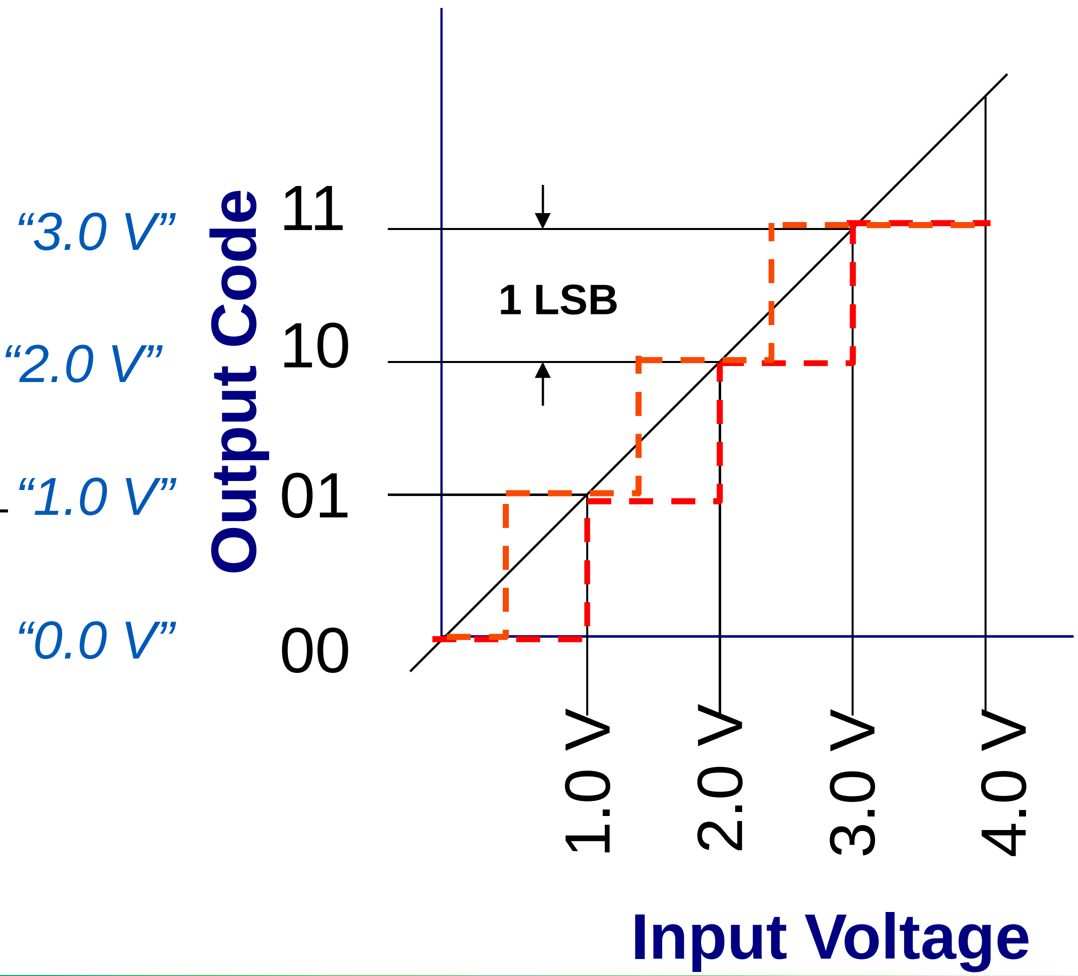
The ADC produces a given output code for all voltages within a specific range

The ideal transfer function A/D converter is a stair-step function.

Consider a 2-bit ADC
- 0 to 4 V input
- LSB = $4/2^2$ = 1 V

Red line
- Truncation
- Maximum error is -1 LSB or + 0 LSB

Blue line
- Rounding
- Maximum error in conversion is usually ± 1/2 bit.
- *Half as much error if we limit range to $V_{ref}(1-2^N/2)$*



Transfer Function Equation

General Equation

n = converted code

V_{in} = sampled input voltage

V_{+ref} = upper end of input voltage range

V_{-ref} = lower end of input voltage range

N = number of bits of resolution in ADC

$$n = \left\lfloor \frac{(V_{in} - V_{-ref})\,2^N}{V_{+ref} - V_{-ref}} + 1/2 \right\rfloor$$

Simplification with V-ref = 0 V

$$n = \left\lfloor \frac{(V_{in})\,2^N}{V_{+ref}} + 1/2 \right\rfloor$$

$$n = \left\lfloor \frac{3.30v\ 2^{10}}{5v} + 1/2 \right\rfloor = 676$$

$\lfloor X \rfloor = I$ ***floor function: nearest integer I such that I <= X***

Example

Your voltage range is 3.3 to 0 V, device is an 8-bit ADC

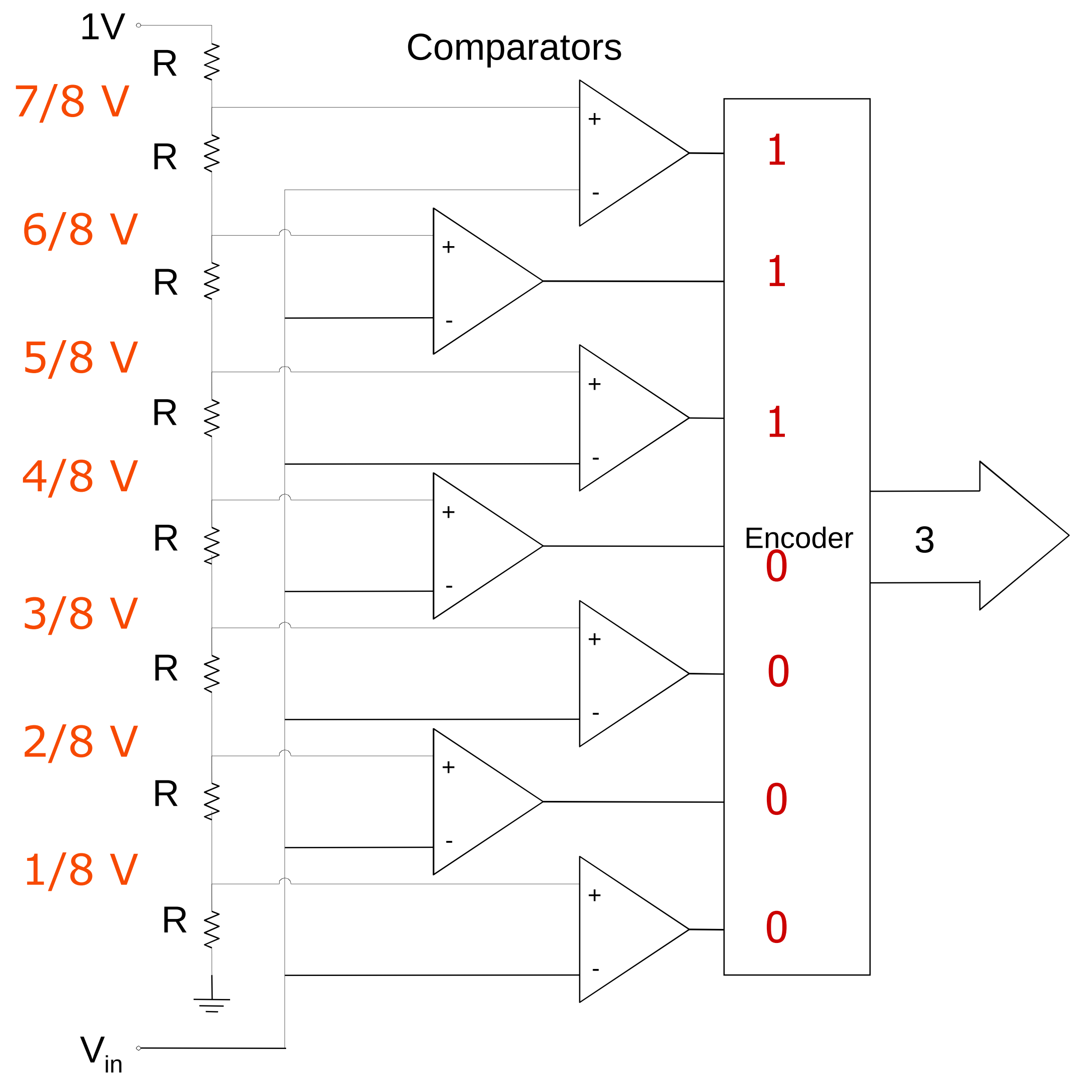
a) What is the step size?

b) If v_{in} is 0.9v, what is n?

$$n = \left\lfloor \frac{(V_{in} - V_{-ref})2^N}{V_{+ref} - V_{-ref}} + 1/2 \right\rfloor$$

A/D – Flash Conversion

- A multi-level voltage divider is used to set voltage levels over the complete range of conversion.
- A comparator is used at each level to determine whether the voltage is lower or higher than the level.
- The series of comparator outputs are encoded to a binary number in digital logic (a priority encoder)
- Components used
  - 2^N resistors
  - 2^N-1 comparators
- Note
  - This particular resistor divider generates voltages which are *not* offset by ½ bit, so maximum error is 1 bit
  - We could change this offset voltage by using resistors of values R, 2R, 2R ... 2R, 3R (starting at bottom)



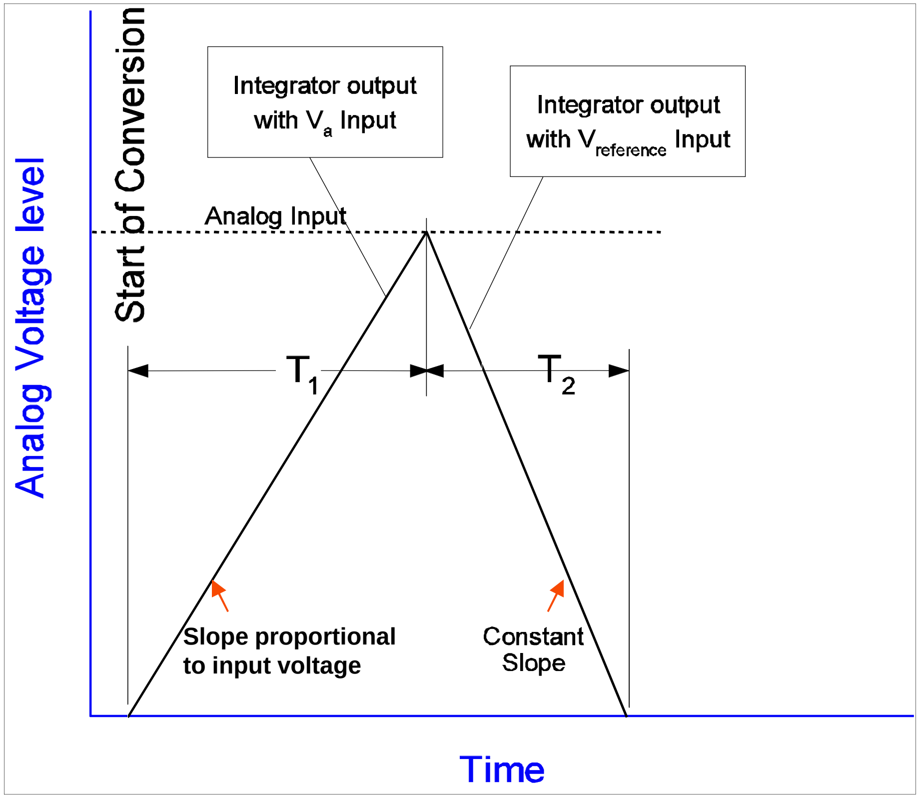
ADC - Dual Slope Integrating

Operation

- Input signal is integrated for a fixed time
- Input is switched to the negative reference and the negative reference is then integrated until the integrator output is zero
- The time required to integrate the signal back to zero is used to compute the value of the signal
- Accuracy dependent on V_{ref} and timing

Characteristics

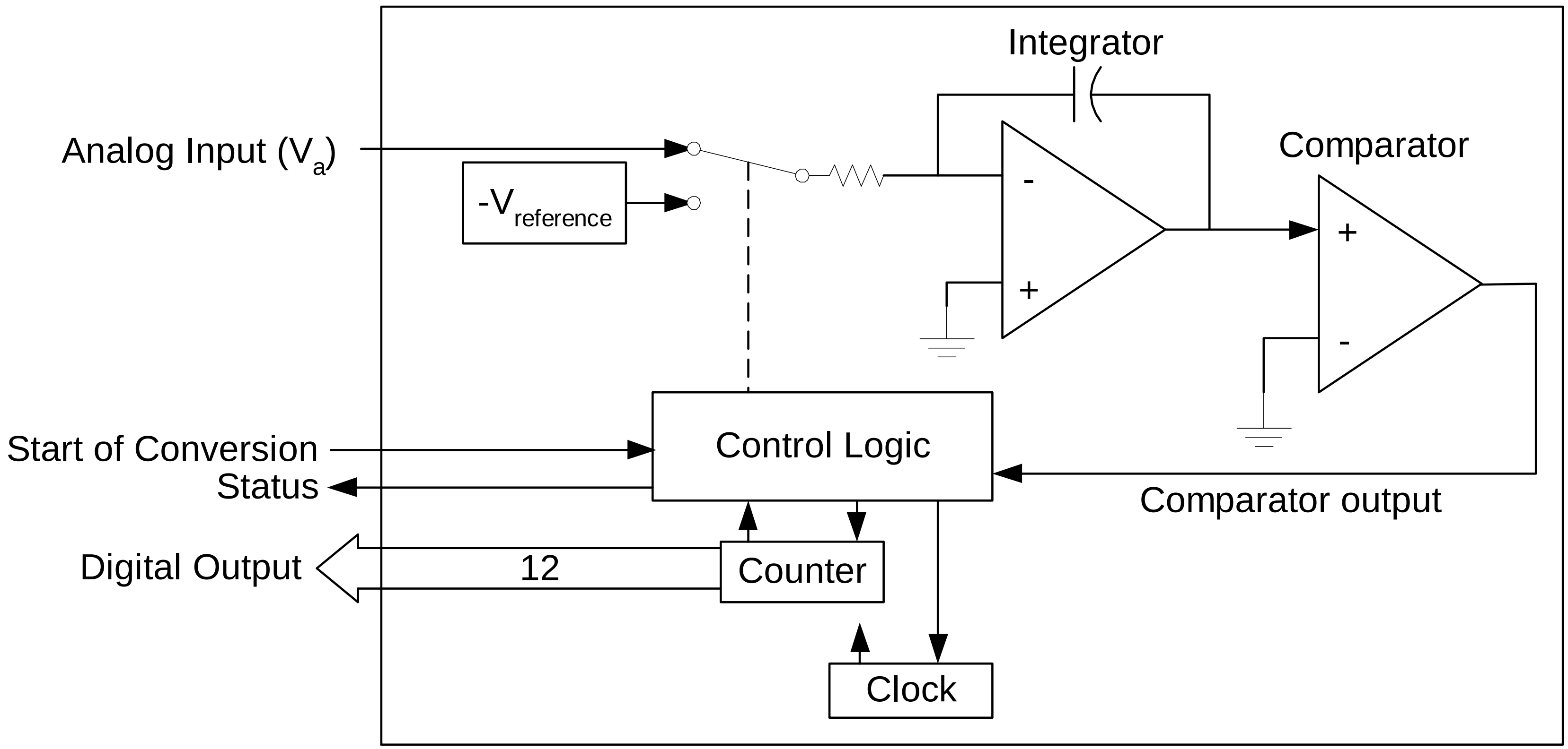
- Noise tolerant (Integrates variations in the input signal during the T_1 phase)
- Typically slow conversion rates (Hz to few kHz)



$$\frac{1}{C}\int_0^{T_1} V_{in}\,dt = -\frac{1}{C}\int_0^{T_2} V_{ref}\,dt$$

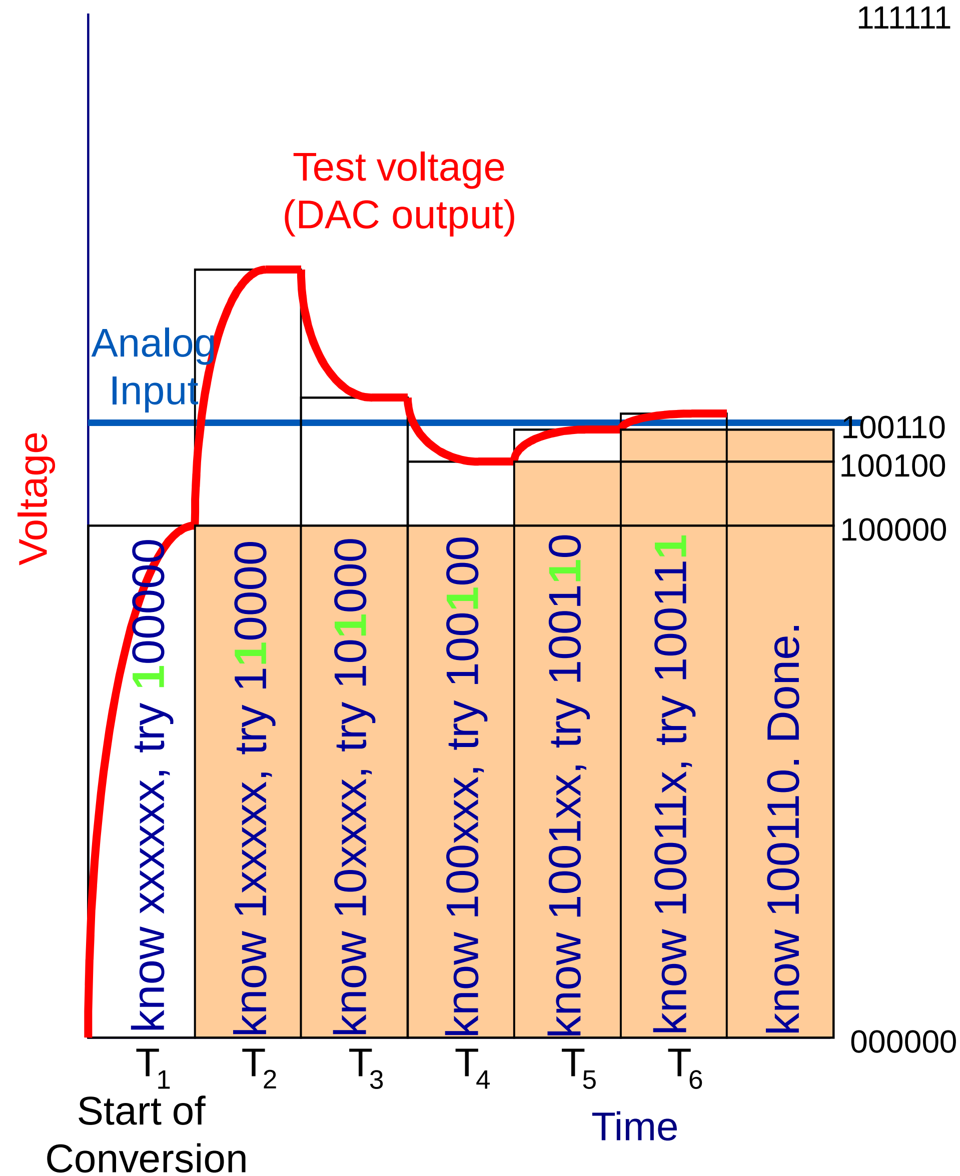
$$V_{in} = V_{ref}\frac{T_2}{T_1}$$

ADC - Dual Slope Integrating



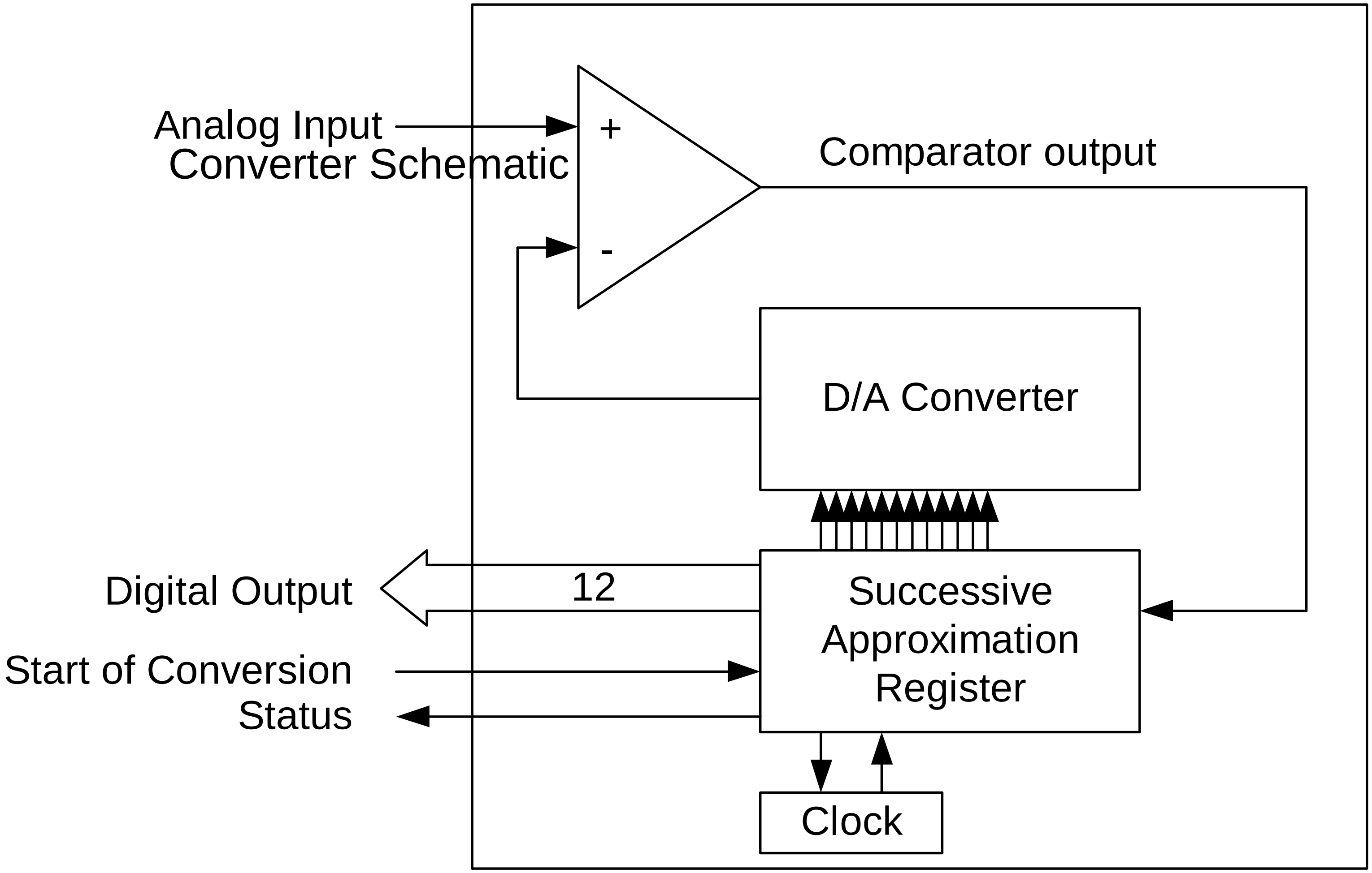
ADC - Successive Approximation Conversion

- Successively approximate input voltage by using a binary search and a DAC
- SA Register holds current approximation of result
- Set all DAC input bits to 0
- Start with DAC's most significant bit
- Repeat
  - Set next input bit for DAC to 1
  - Wait for DAC and comparator to stabilize
  - If the DAC output (test voltage) is **smaller** than the input then set the current bit to 1, else clear the current bit to 0



A/D - Successive Approximation

Converter Schematic



ADC Performance Metrics

Linearity measures how well the transition voltages lie on a straight line.

Differential linearity measure the equality of the step size.

Conversion time:between start of conversion and generation of result

Conversion *rate* = inverse of conversion *time*

Sampling Problems

Nyquist criterion

- $F_{sample} >= 2 * F_{max\ frequency\ component}$
- Frequency components above ½ F_{sample} are aliased, distort measured signal

Nyquist and the real world

- This theorem assumes we have a perfect filter with “brick wall” roll-off
- Real world filters have more gentle roll-off
- Inexpensive filters are even worse (e.g. first order filter is 20 dB/decade, aka 6 dB/octave)
- So we have to choose a sampling frequency high enough that our filter attenuates aliasing components adequately

Quantization

Quantization: converting an analog value (infinite resolution or range) to a digital value of N bits(finite resolution, 2^N levels can be represented)

Quantization error

- Due to limited resolution of digital representation
- <= $1/(2*2^N)$
- Acoustic impact can be minimized by dithering (adding noise to input signal)

16 bits…. too much for a generic microcontroller application?

- Consider a 0-5V analog signal to be quantized
- The LSB represents a change of 76 microvolts
- Unless you're very careful with your circuit design, you can expect noise of of at least tens of millivolts to be added in
- 10 mV noise = 131 quantization levels. So $\log_2 131 = 7.03$ bits of 16 are useless!

Inputs

Multiplexing

- Typically share a single ADC among multiple inputs
- Need to select an input, allow time to settle before sampling

Signal Conditioning

- Amplify and filter input signal
- Protect against out-of-range inputs with clamping diodes

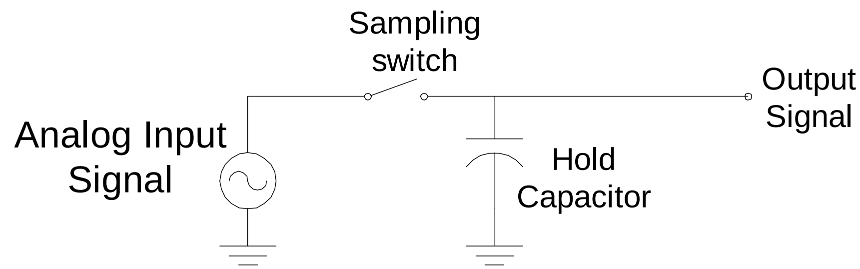
Sample and Hold Devices

Some A/D converters require the input analog signal to be held constant during conversion, (eg. successive approximation devices)

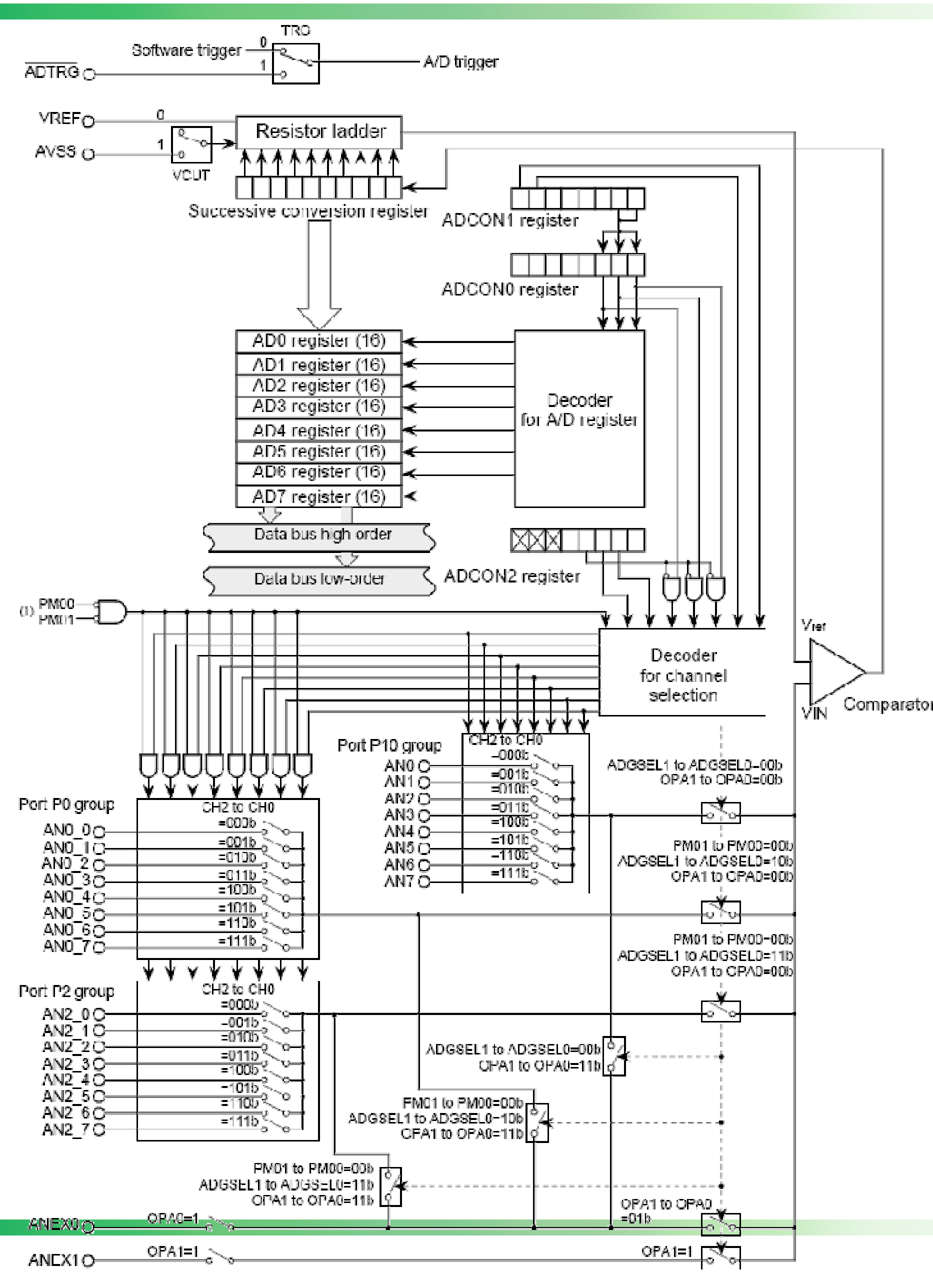
In other cases, peak capture or sampling at a specific point in time necessitates a sampling device.

This function is accomplished by a sample and hold device as shown to the right:

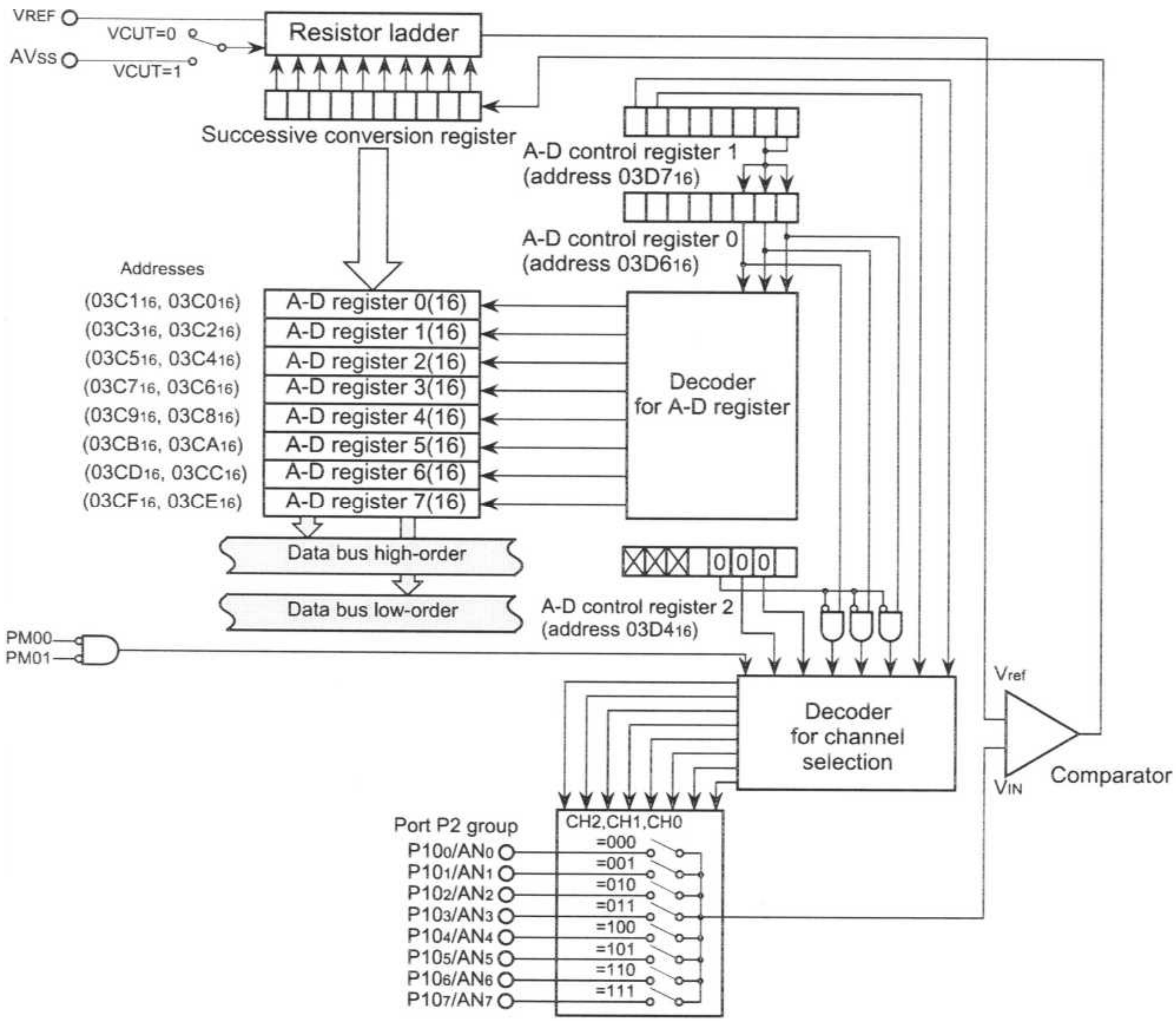
These devices are incorporated into some A/D converters



M30626 A/D Converter Overview



M30262 Converter Overview (626P similar)



Digital to Analog Conversion

May need to generate an analog voltage or current as an output signal

- Audio, motor speed control, LED brightness, etc.

Digital to Analog Converter equation

- n = input code
- N = number of bits of resolution of converter
- V_{ref} = reference voltage
- V_{out} = output voltage
- $V_{out} = V_{ref} * n/(2^N)$