جامعة المستقبل

نحو جامعة مستدامة

Al-Mustaqbal University - College of engineering
Department of computer engineering

Second stage

Lecture Week 13

"Analog-to-Digital Converter (ADC)"

By DR. Mohammed Kadhim Rahma

2025

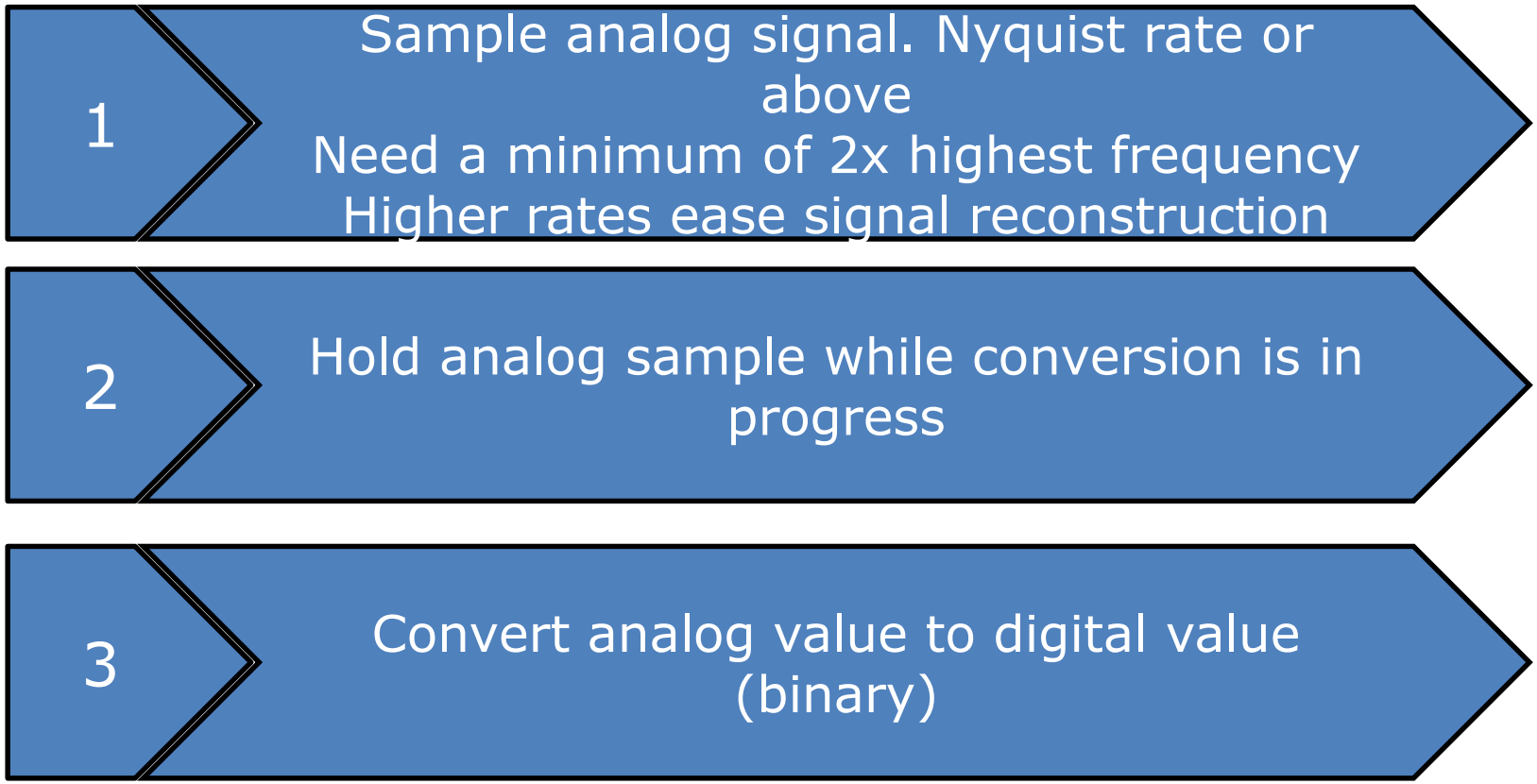
Learning Objectives

After this Lecture you will be able to:

- Determine the resolution and accuracy of a digitized analog signal.
- Explain how analog-to-digital converts operate
- Compare and contrast the characteristics several commonly used analog-to-digital converts operate

Analog-to-Digital Conversion (ADC)

Converting continuous signals to digital values requires 3 steps



Types of Analog-to-digital Converters

Integrating

- High Accuracy
- Low Speed
- Low Cost
- Not commonly used in DAQ

Tracking (Counter Type)

- High Speed in tracking mode
- Slow Conversion times (Some Sub-types)
- Noise Sensitive

Successive Approximation

- Conversion time independent of input value
- Most commonly used in DAQ applications

Types of Analog-to-digital Converters

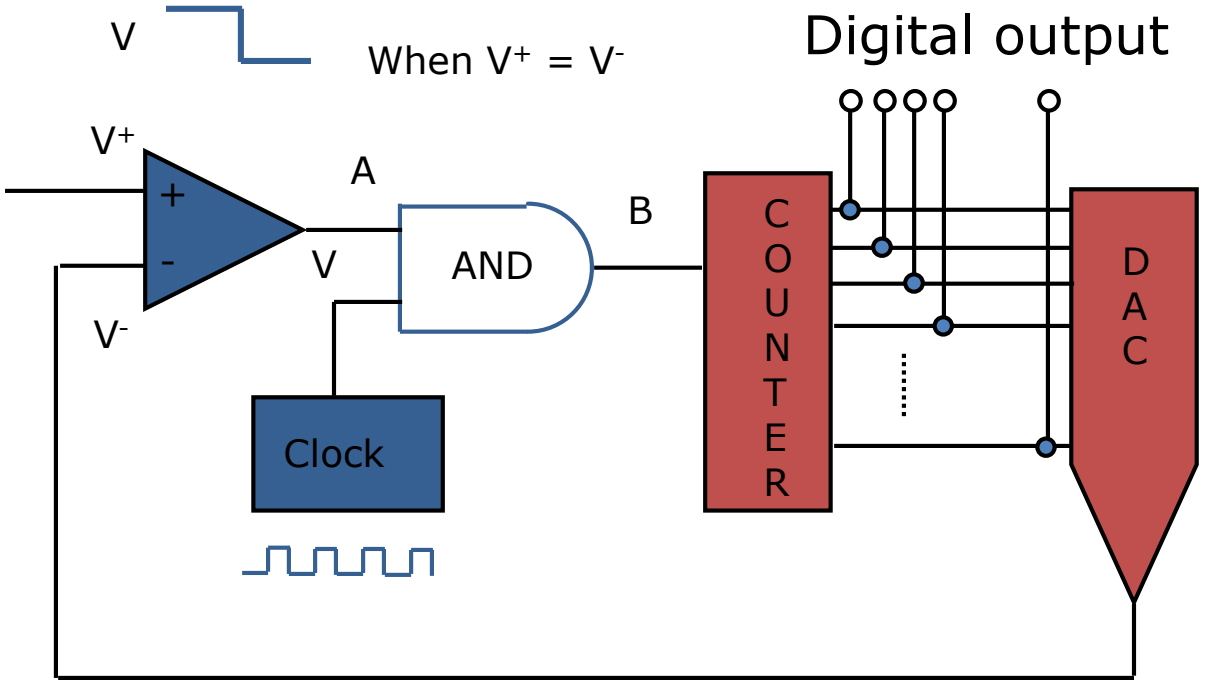
"Flash" or Parallel

- Multi-Comparators
- Highest Speed
- High Cost

Delta/Sigma

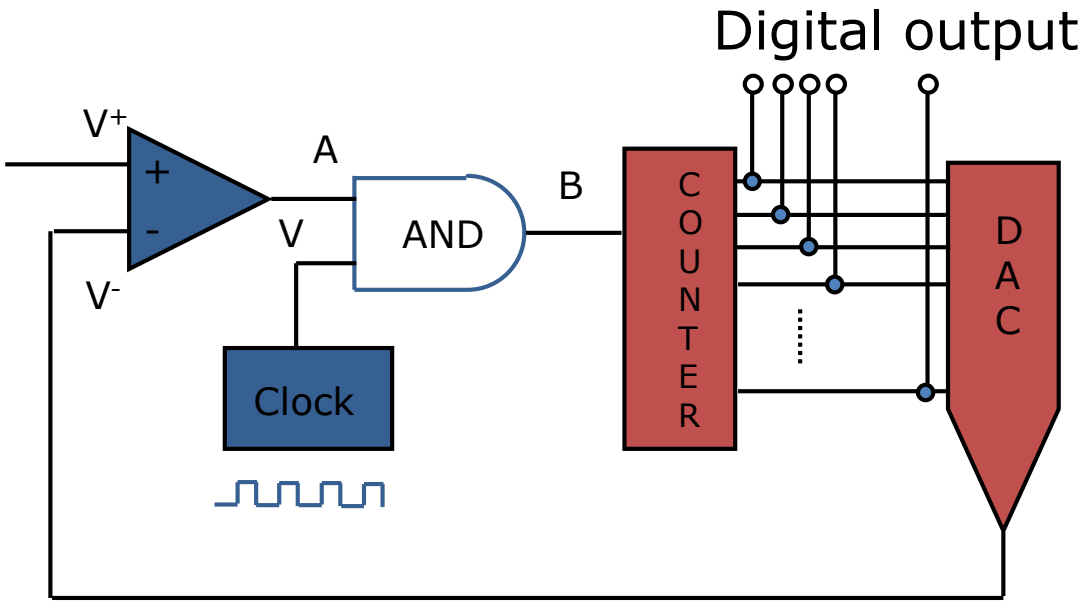
- One bit Conversion
- High Resolution
- Ratio of 1-to-0 represent input
- Uses digital filtering

Counter-Type Analog-to-Digital Converter Operation



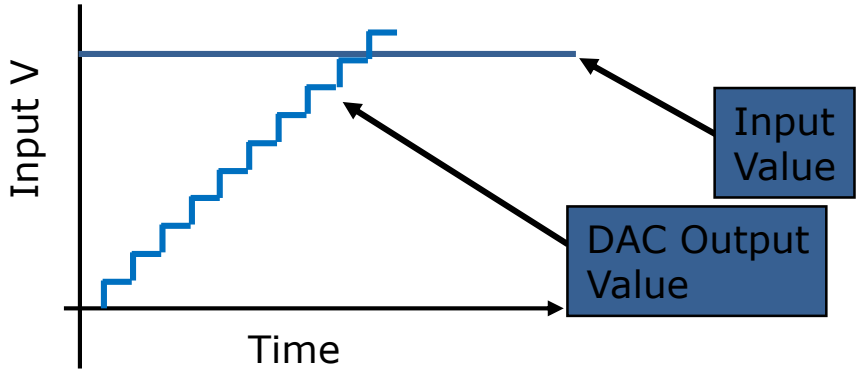
1.) Input a constant value. Requires a sample and hold circuit. (Not Shown)
2.) AND gate passes clock signal when point A logic high
3.) Counter incremented by signal B
4.) DAC output increases as counter output increases

Counter-Type Analog-to-Digital Converter Operation



Tracking A/D converters use up/down counters to minimize conversion times

5.) Counter stops when DAC V exceeds input V



Conversion time depends on the input level

Successive Approximation ADC

Counter A/D converters conversion time proportional to the input level. Improve conversion speed using a binary search technique.

Procedure

1. Set MSB to 1

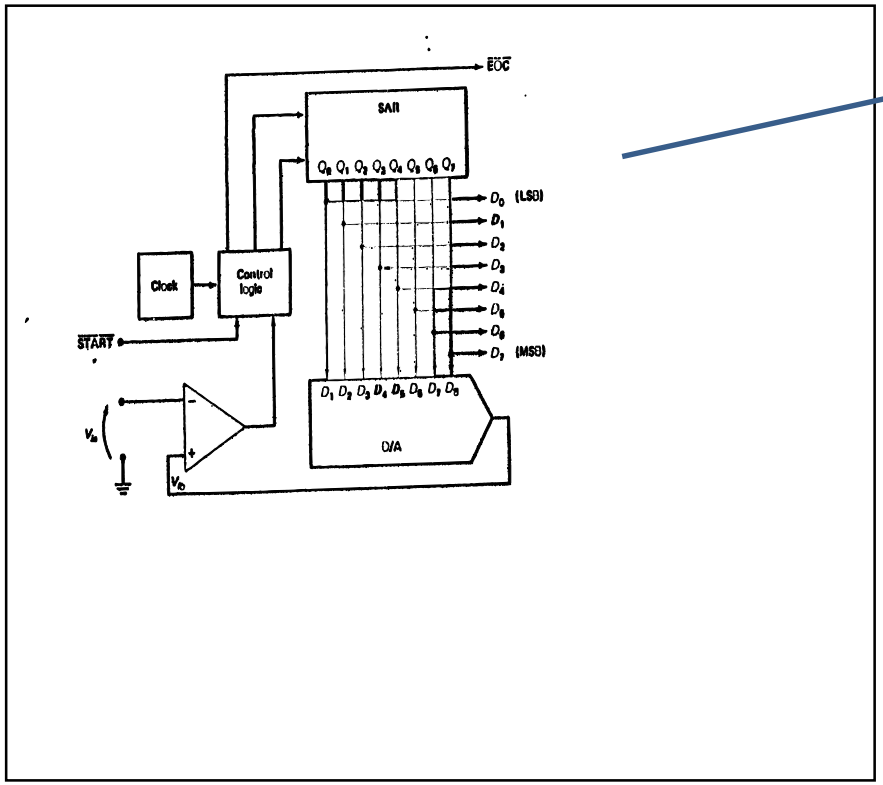
2. Test input, V_{in}, against DAC output, V_{DAC}

3. If $V_{DAC} > V_{in}$, reset bit to 0, else bit = 1 ($V_{DAC} < V_{in}$)

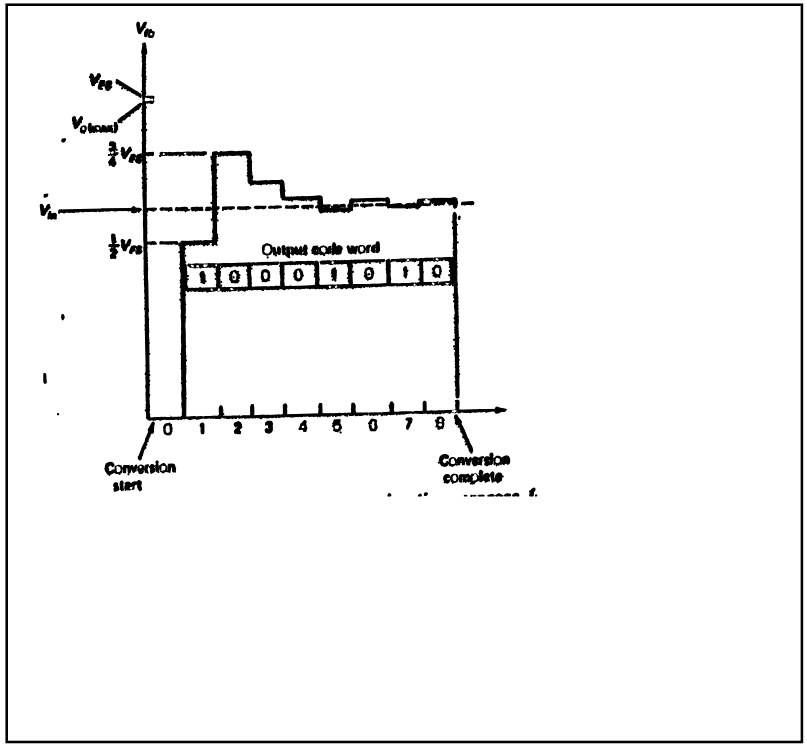
4. Move to next bit and repeat steps 1 - 3

The input is converted to digital value in n steps, where n = the number of bits in digital representation

Successive Approximation ADCs



Successive Approx. Register



Example: An 8-bit successive approximation ADC has an input voltage of 13.478 V. The ADC full scale input voltage is 20 V. Use the successive approximation algorithm given previously to determine the binary value. Assuming that each bit test takes a single clock cycle, determine the maximum conversion time for the ADC if it is clocked at 4.77 MHz.

Example Solution

Voltage Weight	10.0	5.0	2.5	1.25	0.625	0.3125	0.15625	0.078125	Cycle
Bit	D7	D	D5	D4	D3	D2	D1	D0	
Set D7	0	0	0	0	0	0	0	0	
Set D6	1	0	0	0	0	0	0	0	1
Set D5	1	0	0	0	0	0	0	0	2
Set D4	1	0	1	0	0	0	0	0	3
Set D3	1	0	1	0	0	0	0	0	4
Set D2	1	0	1	0	1	0	0	0	5
Set D1	1	0	1	0	1	1	0	0	6
Set D0	1	0	1	0	1	1	0	0	7
	1	1	1	1	1	1	1	1	8

1. 10<13.478 bit remains set
2. 10+5>13.478 reset bit
3. 10+2.5<13.478 bit remains set
4. 10+2.5+1.25>13.478 reset bit
5. 10+2.5+0.625<13.478 bit remains set
6. 10+2.5+0.625+0.3125<13.478 bit remains set
7. 10+2.5+0.625+0.3125+.15625>13.478 bit reset
8. 10+2.5+0.625+0.3125+.15625+0.078125>13.478 bit reset

Example Solution (Continued)

Summary Results

Final binary value: $B=10101100_2$

Final voltage: 10+2.5+0.625+0.3125 = 13.4375 V

Quantization Error voltage: V_{QE}=13.478 V-13.4375 V = 0.0405 V or 40.5 mV

Determine conversion time

Define T_c as clock period = $1/f_c$ Where f_c = ADC Clock = 4.77 MHz

$$T_c = \frac{1}{f_c} = \frac{1}{4.77\times10^6\,Hz} = 2.096\times10^{-7}\ S = 0.2096\ \mu S$$

All conversions take 8 clock cycles so maximum conversion time is $8\cdot T_c$

$T_{con} = 8\cdot T_c = 8\cdot(0.2096\ \mu S) =$ **1.667 μS** Ans

Example Solution (Continued)

What is the highest frequency that the system can convert without folding or aliasing. Assume that the sample and hold time is zero.

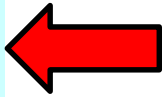
Define the maximum conversion rate frequency, $f_{con(max)}$

$$f_{con(max)} = \frac{1}{T_{con}} = \frac{1}{1.677 \times 10^{-6} S} = 596{,}250 \text{ Hz}$$

Signals must be sampled at least twice per period (Nyquist rate)

$$2 \cdot f_{in} = f_{con(max)}$$

$$f_{in} = \frac{f_{con(max)}}{2} = \frac{596{,}250 \text{ Hz}}{2} = 298{,}125 \text{ Hz}$$

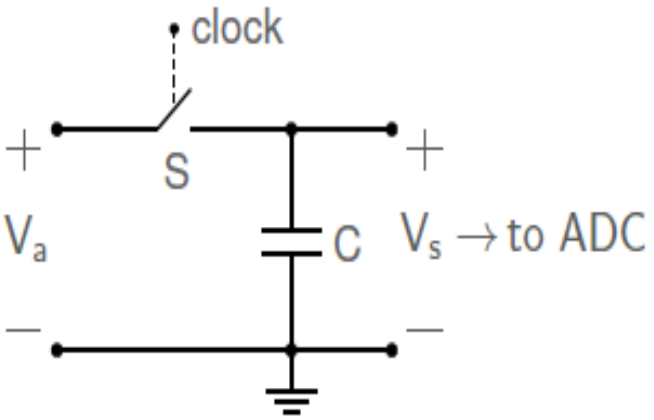


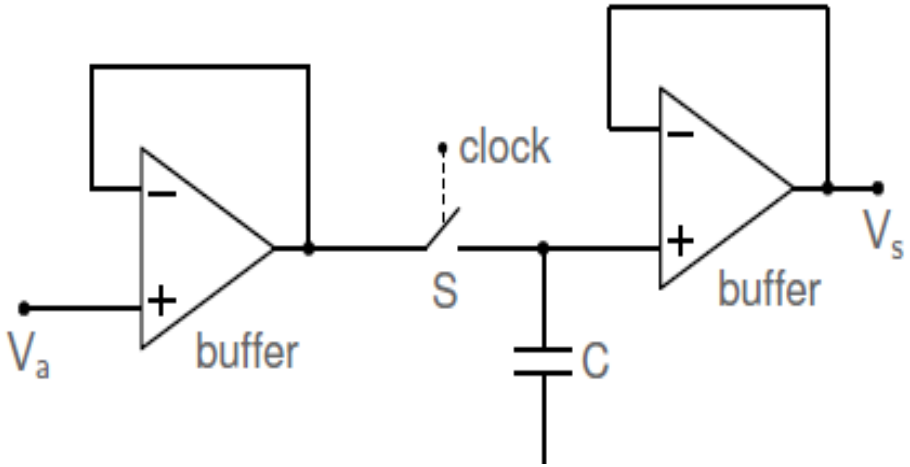
Ans

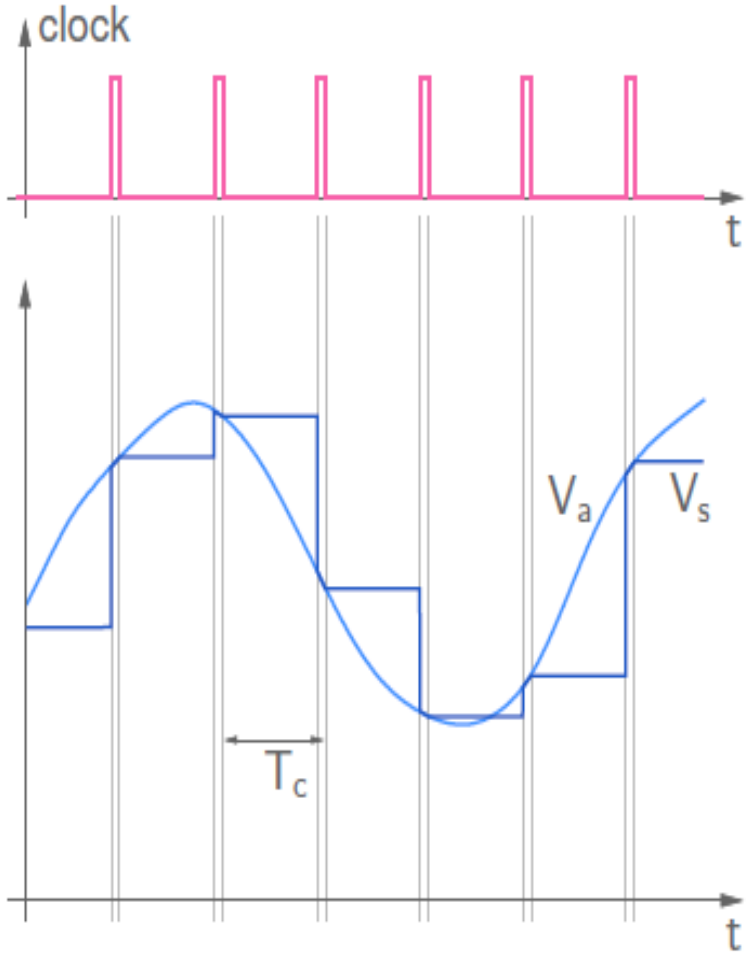
Open Questions

- Is there another way to design low power ADCs?
- Is it recommended to reduce the analog part and put more effort in the digital part?
- How do I achieve a high SNR with low power ADCs?
- Is it better to have only one block with high frequency or many blocks with low frequency?
- How can asynchronous designs help me?
- How do I realize a low power ADC in sub-micron technologies?

ADC: sampling of input signal



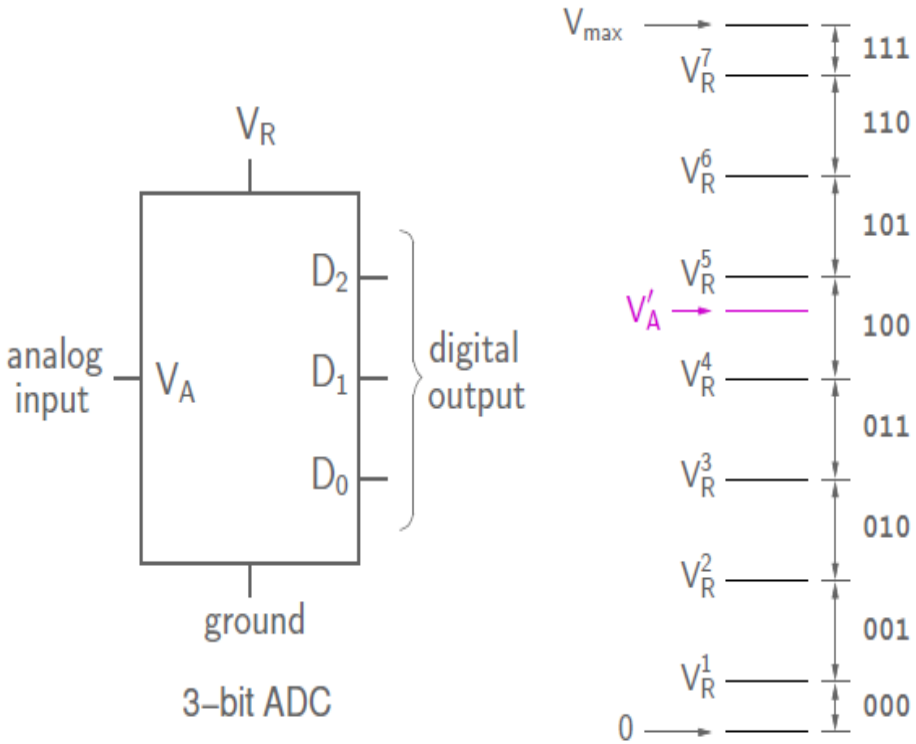




Application of ADC

- ADC are used virtually everywhere where an analog signal has to be processed, stored, or transported in digital form.
- Some examples of ADC usage are digital volt meters, cell phone, thermocouples, and digital oscilloscope.
- Microcontrollers commonly use 8, 10, 12, or16 bit ADCs, our micro controller uses an 8 or 10 bit ADC

Summary

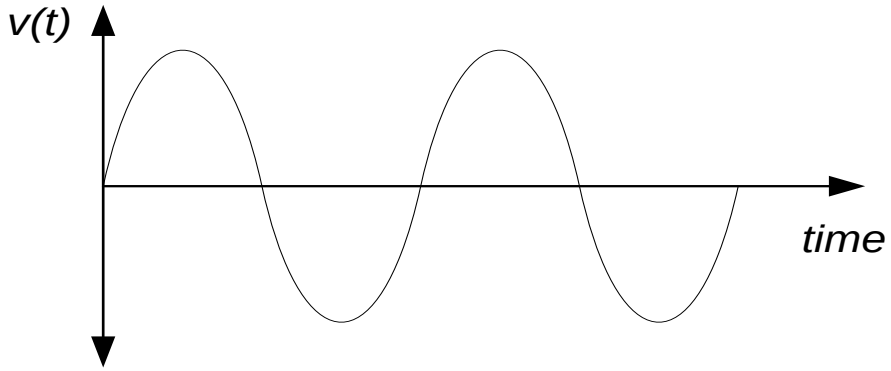


* If the input V_A is in the range $V_R^k < V_A < V_R^{k+1}$, the output is the binary number corresponding to the integer k. For example, for $V_A = V'_A$, the output is 100.
* We may think of each voltage interval (corresponding to 000, 001, etc.) as a "bin." In the above example, the input voltage V'_A falls in the 100 bin; therefore, the output of the ADC would be 100.
* Note that, for an N-bit ADC, there would be 2^N bins.

Signal Reconstruction

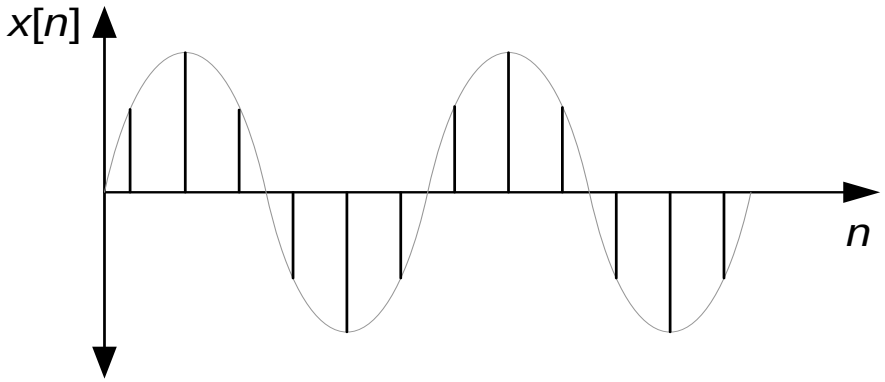
- Continuous time (input signal):

$$V_{RMS\_ct} = \sqrt{\int_{-T/2}^{T/2} \frac{v(t)^2}{T} dt}$$



- Discrete (reconstructed by ADC):

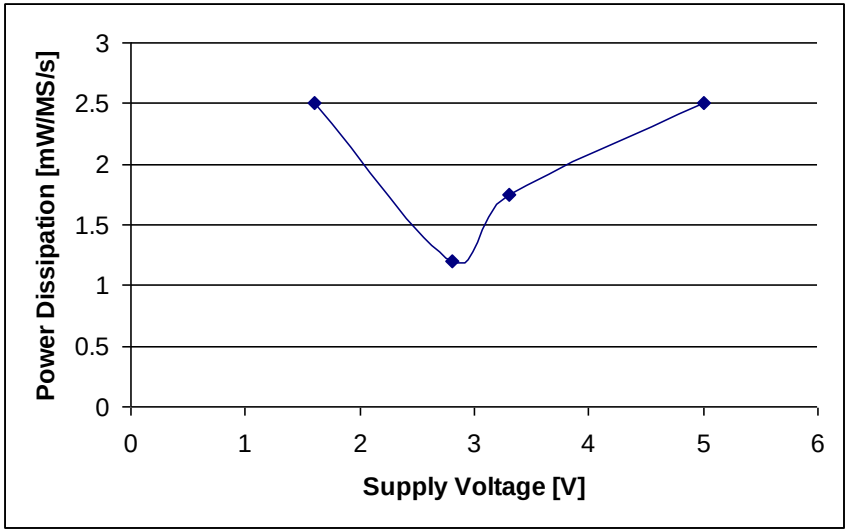
$$V_{RMS\_discrete} = \sqrt{\frac{\sum_{i=0}^{n} x[n]^2}{n}}$$



RMS: root mean square

Voltage supply reduction [Tan00]

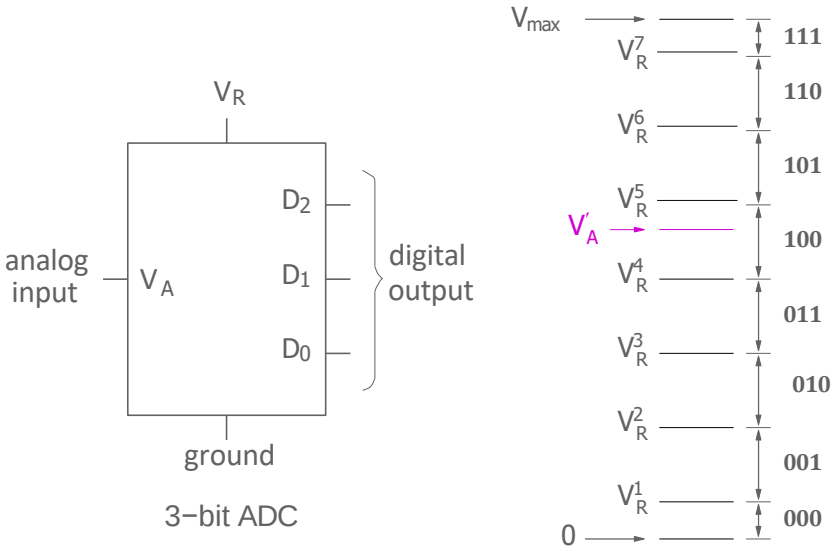
- For analog design, it is shown that a voltage supply reduction does not always lead to a power consumption reduction for several reasons:
  - Threshold of MOS transistors.
  - Loss of maximal amplitudes (SNR degradation).
  - Limits of conduction in analog switches.
  - Low speed of MOS transistors.
  - Limited stack of transistors.



Power consumption of 10-bit S-C 1.5 bit/stage pipelined ADCs in function of the voltage supply.

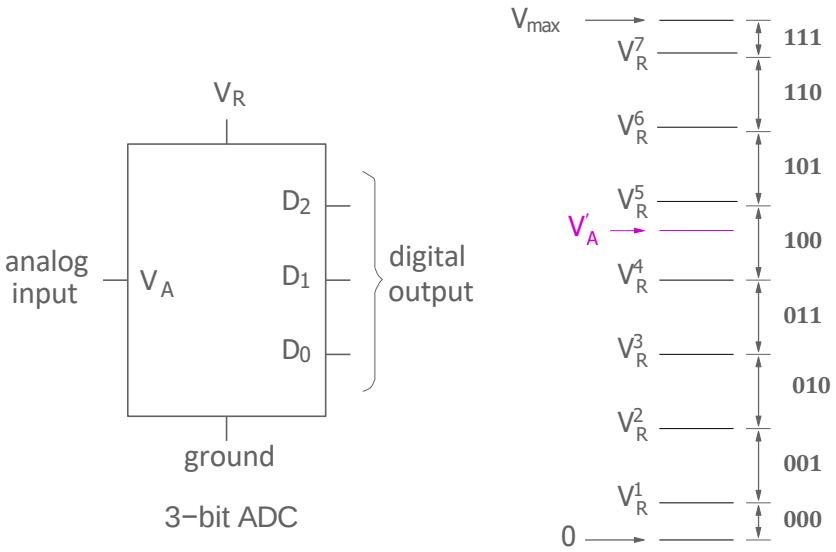
[Tan00] S. Tanner, *Low-power architectures for single-chip digital image sensors*, dissertation, University of Neuchatel, Switzerland, 2000.

ADC: introduction



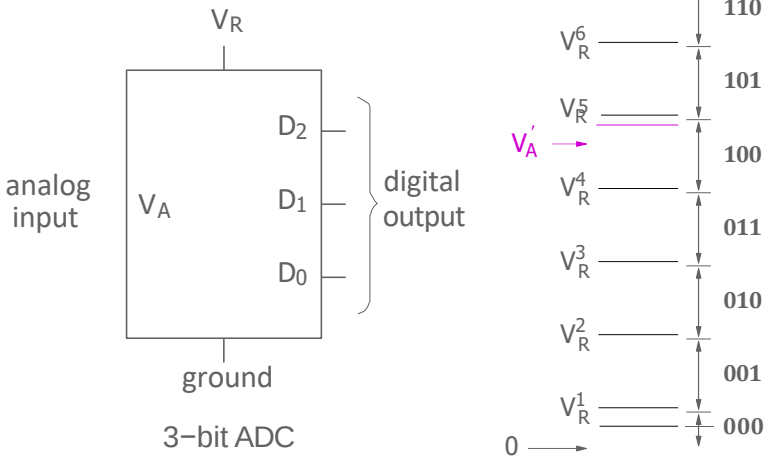
* If the input V_A is in the range $V_R^k < V_A < V_R^{k+1}$, the output is the binary number corresponding to the integer k. For example, for $V_A = V'_A$, the output is 100.
* We may think of each voltage interval (corresponding to 000, 001, etc.) as a "bin." In the above example, the input voltage V'_A falls in the 100 bin; therefore, the output of the ADC would be 100.
* Note that, for an N-bit ADC, there would be 2^N bins.

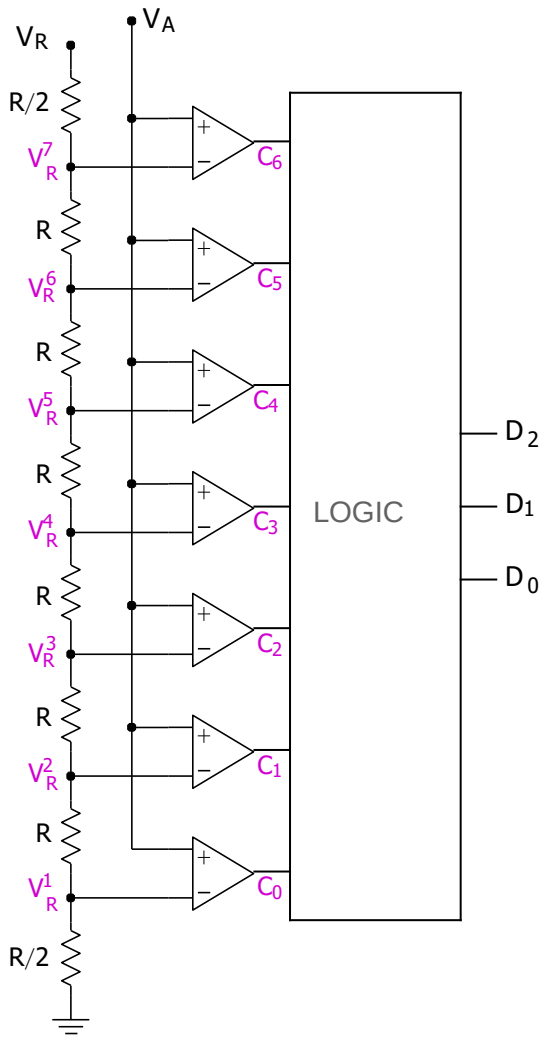
ADC: introduction



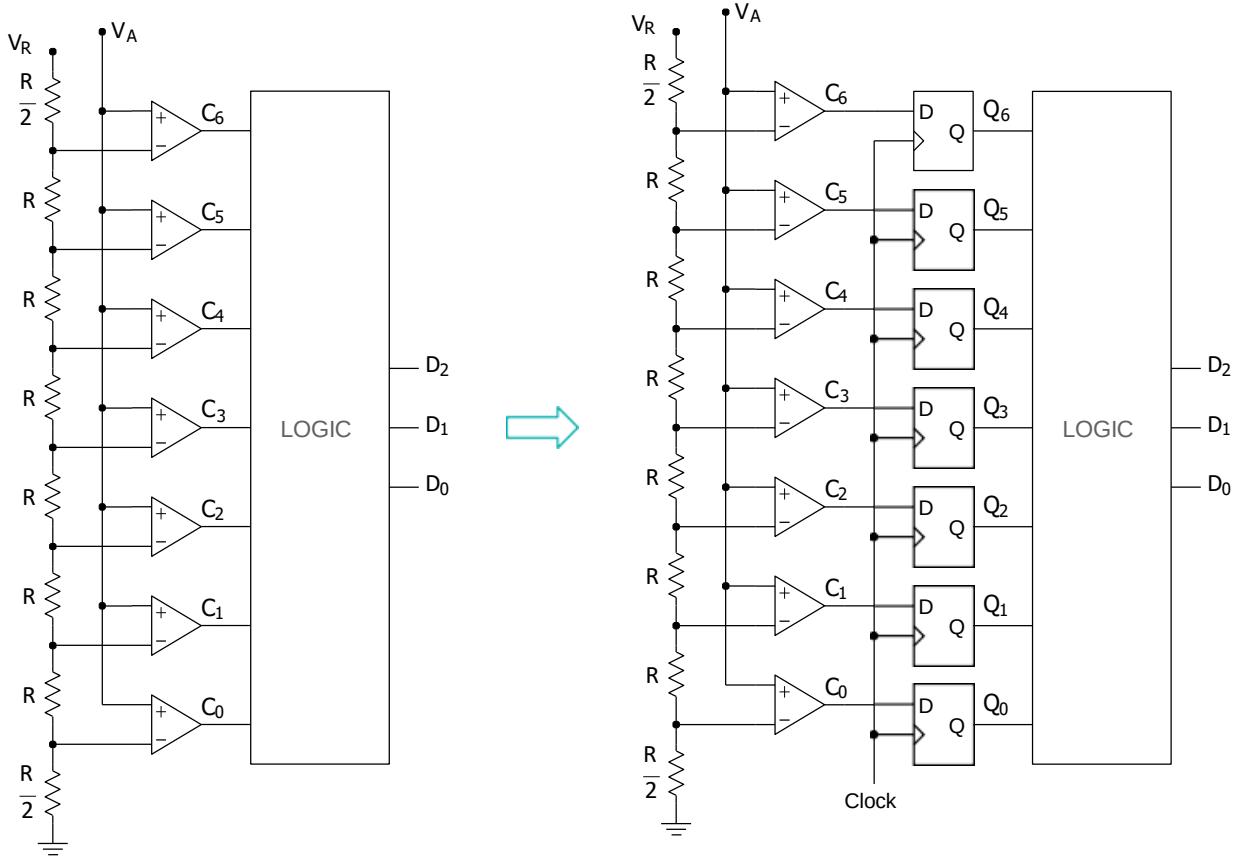
* The basic idea behind an ADC is simple:
  - Generate reference voltages V_R^1, V_R^2, etc.
  - Compare the input V_A with each of V_R^i to figure out which bin it belongs to.
  - If V_A belongs to bin k (i.e., $V_R^k < V_A < V_R^{k+1}$), convert k to the binary format.
* A "parallel" ADC does exactly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

3-bit parallel (flash) ADC



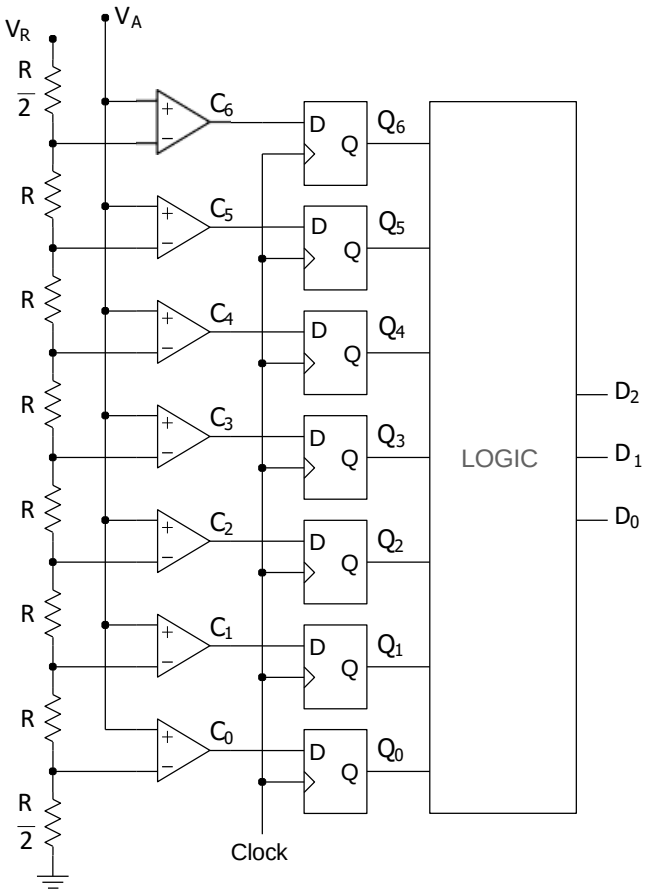


3-bit parallel (flash) ADC



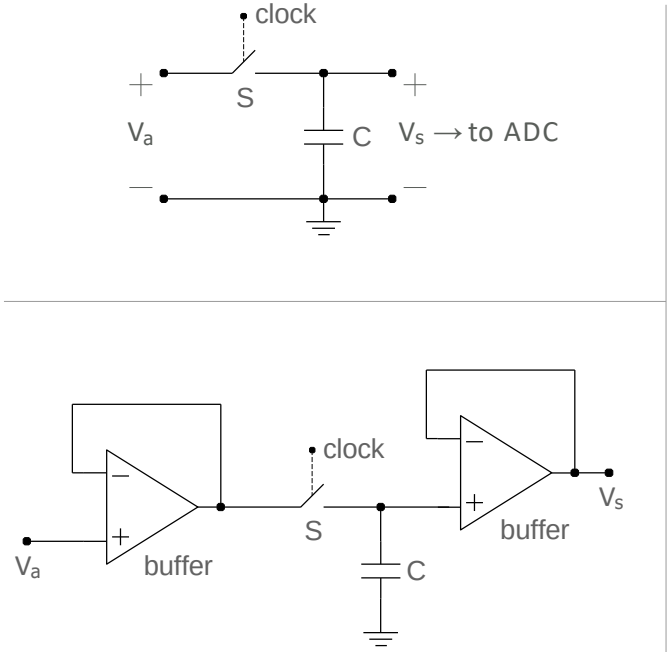
* Practical difficulty: As the input changes, the comparator outputs (C_0, C_1, etc.) may not settle to their new values at the same time. → ADC output will depend on when we sample it.
* Add D flip-flops. Allow sufficient time (between the change in V_A and the active clock edge) so that the comparator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

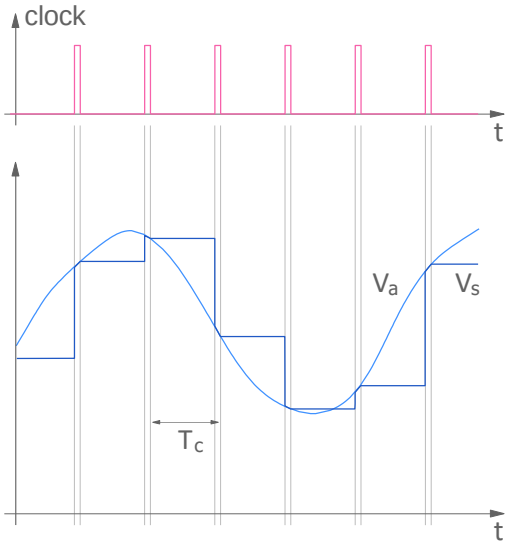
Parallel (flash) ADC



* 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.

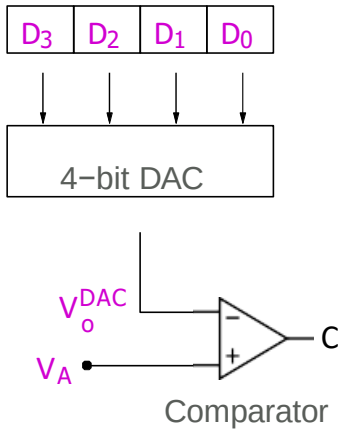
ADC: sampling of input signal





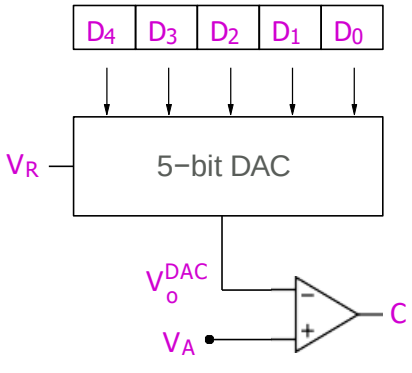
* An ADC typically operates on a "sampled" input signal ($V_s(t)$ in the figure) which is derived from the continuously varying input signal ($V_a(t)$ in the figure) with a "sample-and-hold" (S/H) circuit.
* The S/H circuit samples the input signal $V_a(t)$ at uniform intervals of duration T_c, the clock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

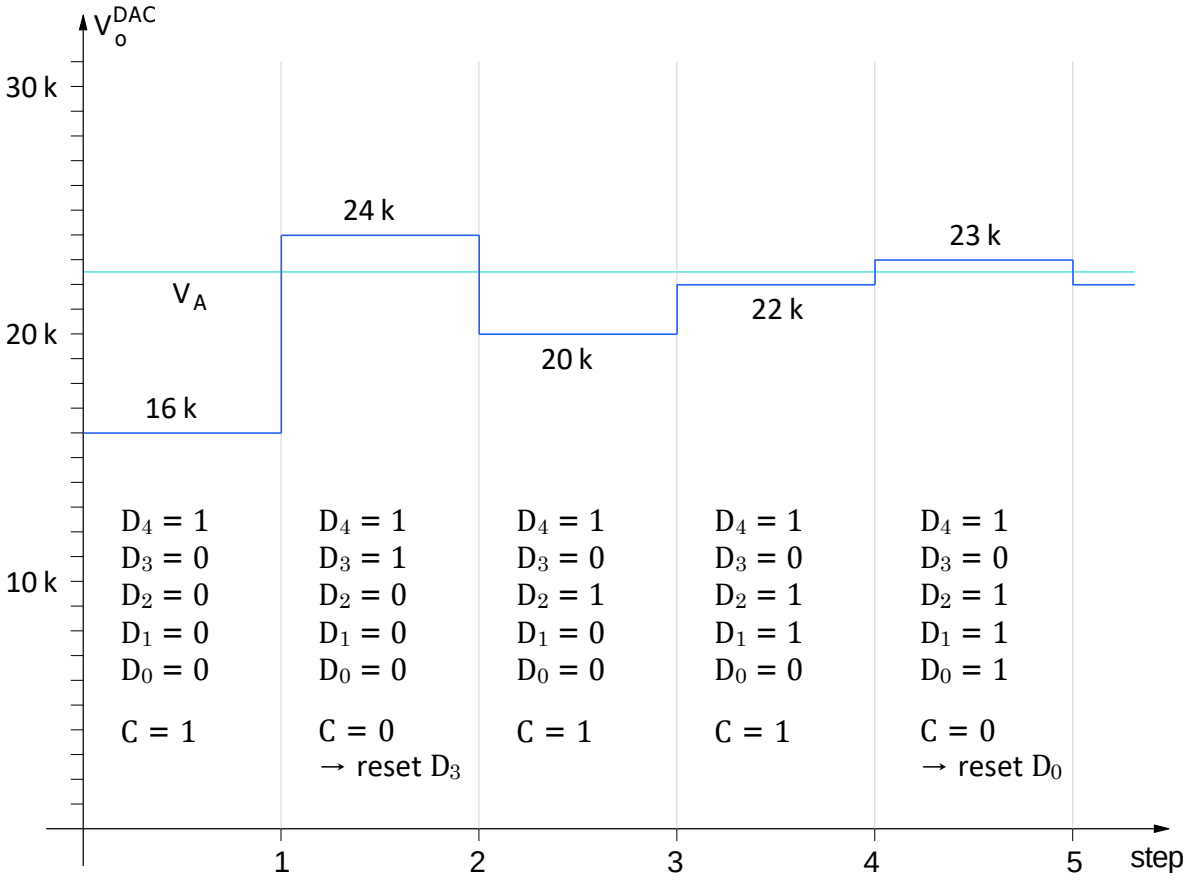


* Suppose we have a 4-bit DAC. We can use it to perform A-to-D conversion by successively setting the four bits as follows.
  - Start with $D_3D_2D_1D_0 = 0000$, $\mathrm{I} = 3$.
  - Set $D[\mathrm{I}] = 1$ (keep other bits unchanged).
  - If $V_o^{DAC} > V_A$ (i.e., $C = 0$), set $D[\mathrm{I}] = 0$; else, keep $D[\mathrm{I}] = 1$.
  - $\mathrm{I} \rightarrow \mathrm{I} - 1$; go to step 1.
* At the end of four steps, the digital output is given by $D_3D_2D_1D_0$.
  Example → next slide.

Successive Approximation ADC

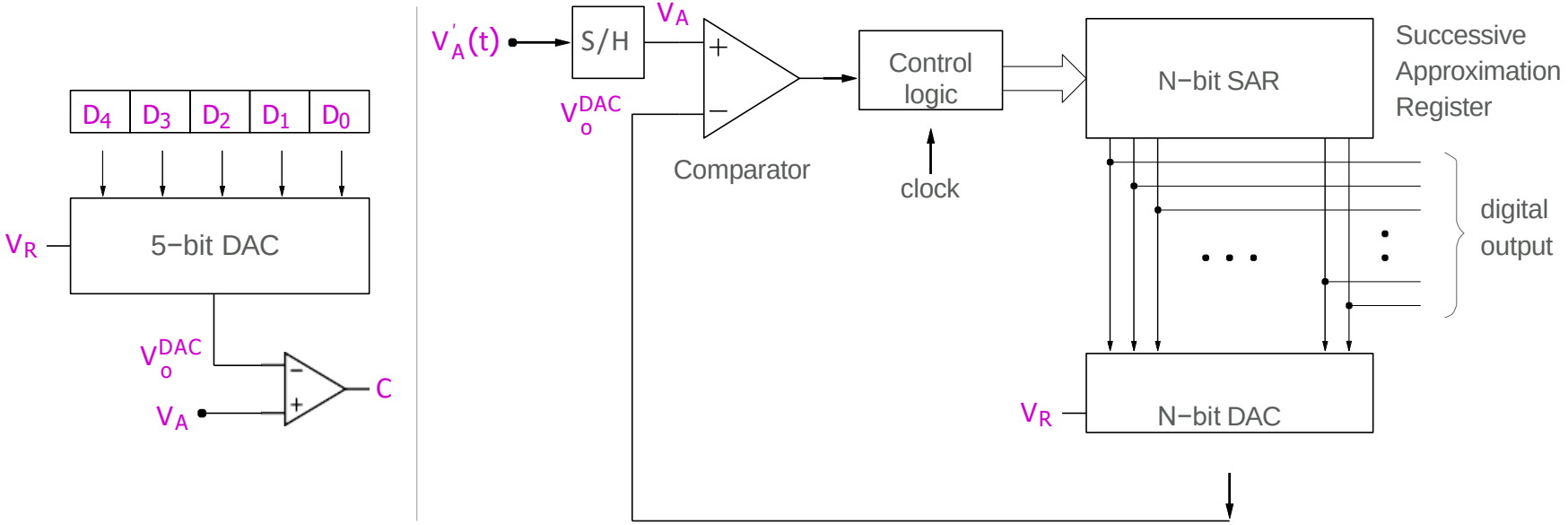


(Note: $k \propto V_R$)



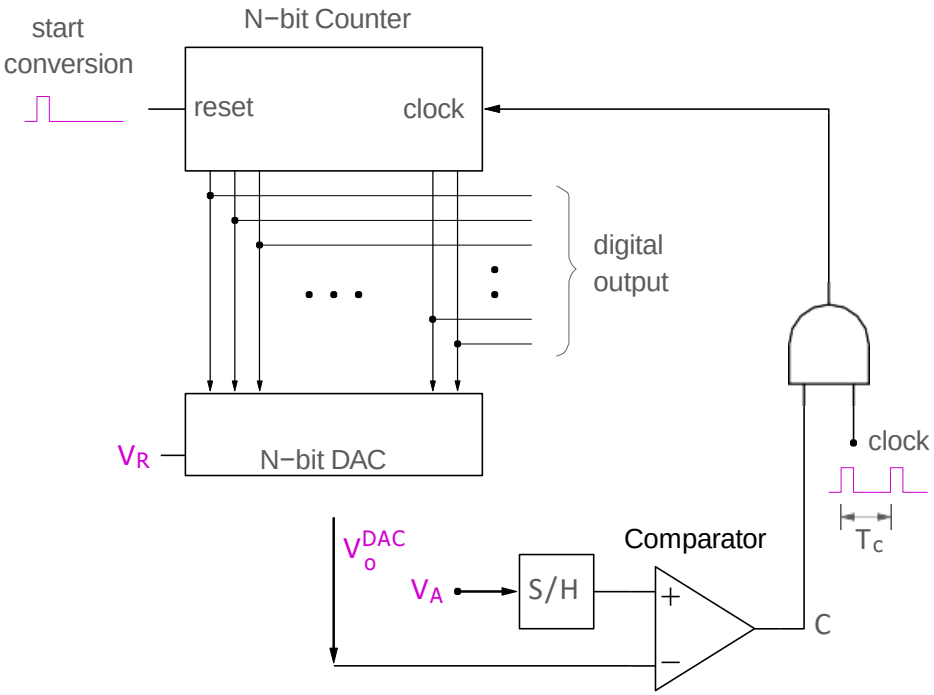
* At the end of the 5th step, we know that the input voltage corresponds to 10110.
* For the digital representation to be accurate up to $\pm\frac{1}{2}$ LSB, ΔV corresponding to $\frac{1}{2}$ LSB is added to V_A (see [Taub]).

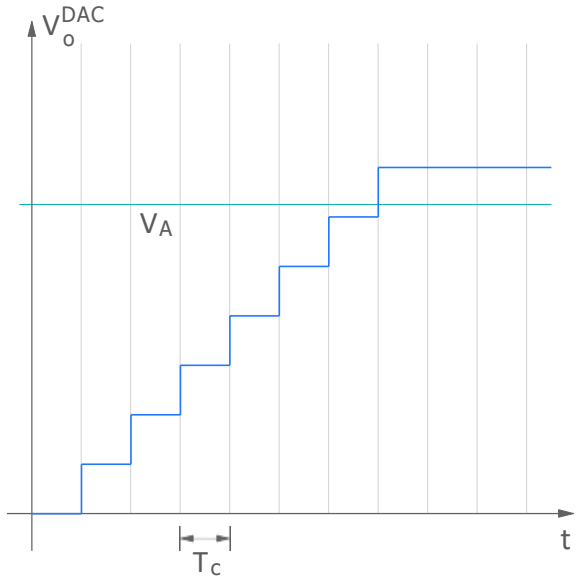
Successive Approximation ADC



* Each step (setting SAR bits, comparison of V_A and V_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 μsec to tens of μsec.
* Useful for medium-speed applications such as speech transmission with PCM.

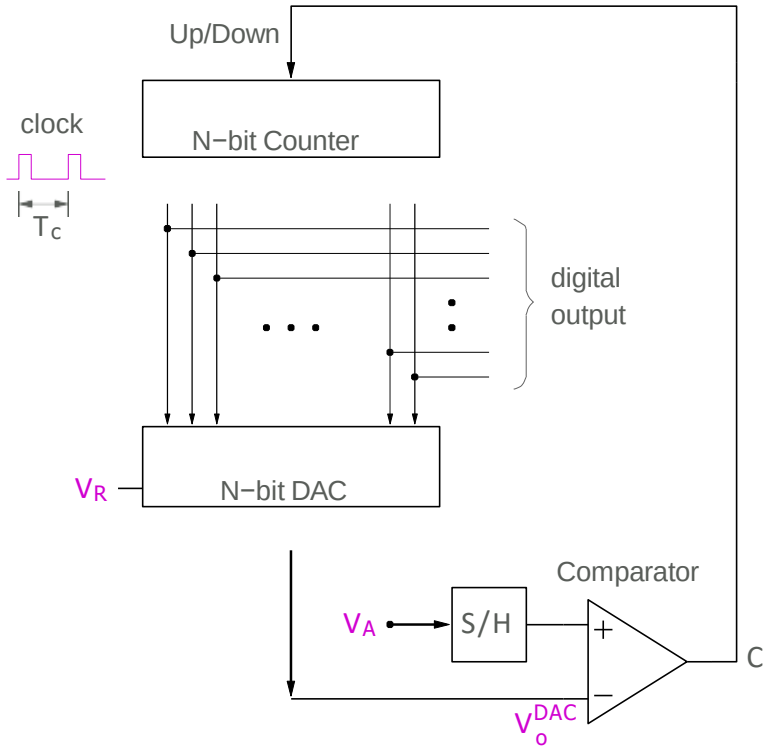
Counting ADC (digital-ramp ADC)

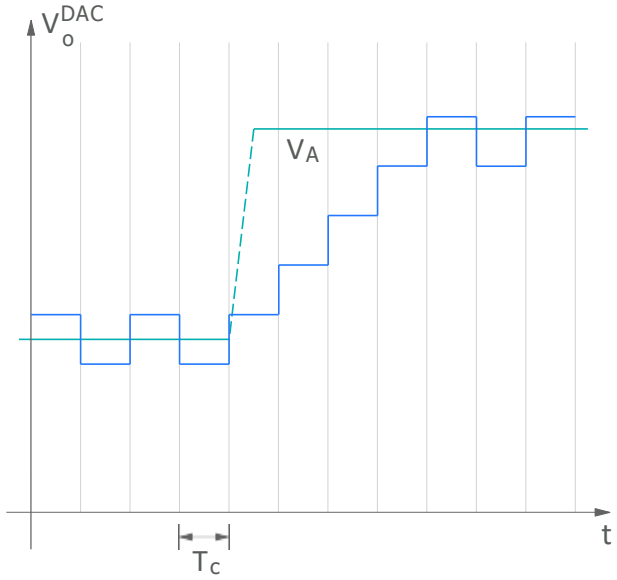




* The "start conversion" signal clears the counter; counting begins, and V_o^{DAC} increases with each clock cycle.
* When V_o^{DAC} exceeds V_A, C becomes 0, and counting stops.
* Simple scheme, but (a) conversion time depends on V_A, (b) slow (takes $(2^N - 1)$ clock cycles in the worst case) → tracking ADC

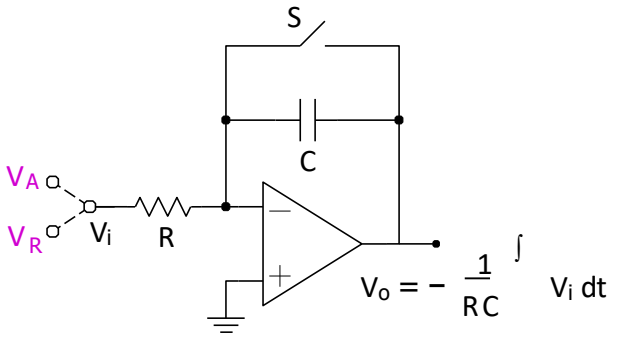
Tracking ADC

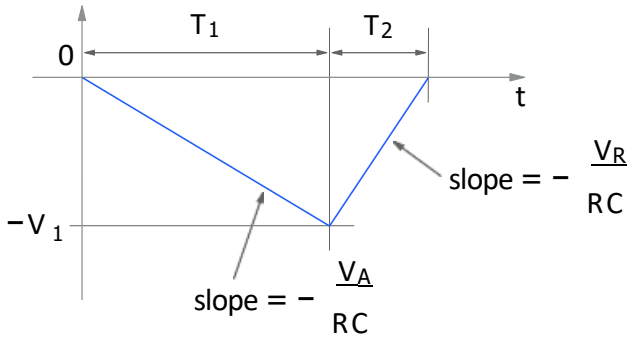




* The counter counts up if $V_o^{DAC} < V_A$; else, it counts down.
* If V_A changes, the counter does not need to start from $000\cdots0$, so the conversion time is less than that required by a counting ADC.
* used in low-cost, low-speed applications, e.g., measuring output from a temperature sensor or a strain gauge

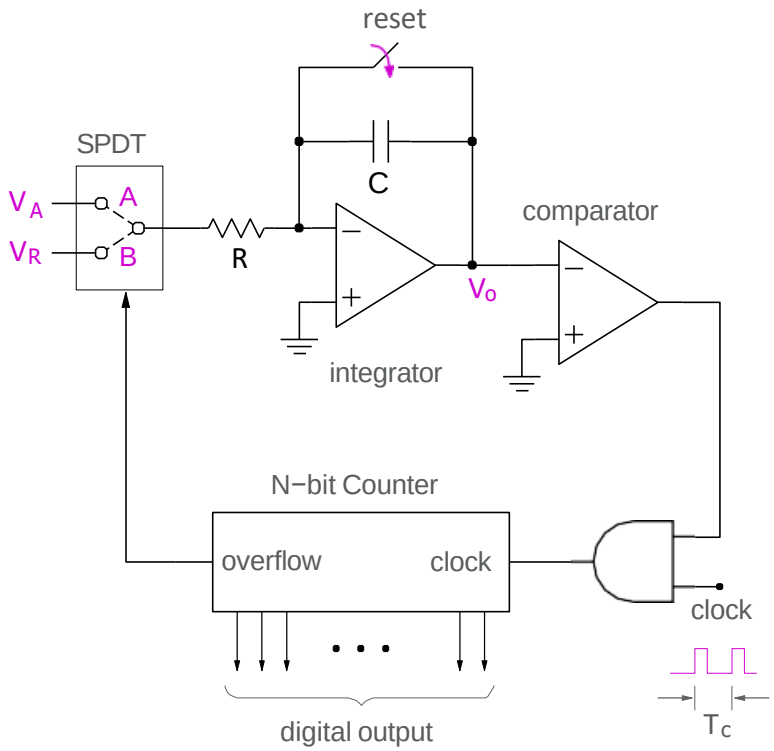
Dual-slope ADC

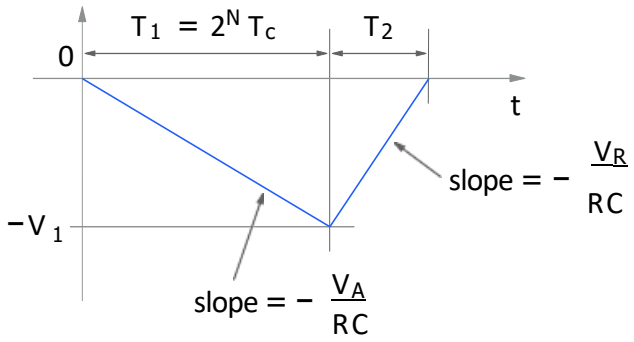




* $t = 0$: reset integrator output V_o to 0 V by closing S momentarily.
* Integrate V_A (voltage to be converted to digital format, assumed to be positive) for a fixed interval T_1.
* At $t = T_1$, integrator output reaches $-V_1 = -V_A \frac{T_1}{RC}$.
* Now apply a reference voltage V_R (assumed to be negative, with $|V_R| > V_A$), and integrate until V_o reaches 0 V.
* 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.
* In the dual-slope ADC, a counter output – which is proportional to T_2 – provides the desired digital output.

Dual-slope ADC





* Start: counter reset to $000\cdots0$, SPDT in position A.
* Counter counts up to 2^N at which point the overflow flag becomes 1, and SPDT switches to position B $\rightarrow$ $T_1 = 2^N T_c$ where T_c is the clock period.
* The counter starts counting again from $000\cdots0$, and stops counting when V_o crosses 0 V. The counter output gives T_2 in binary format.

THANK YOU ☺

UOMU022021 Digital Systems

Department of Computer Engineering

END LESSON 13: ANALOG-TO-DIGITAL CONVERTER (ADC)