Embedded Systems Hardware Interfacing

Notes from AN2834

Norman McEntire

life.augmented

AN2834
Application note

How to get the best ADC accuracy in STM32 microcontrollers

My Own Preface

ADC is Hard (compared to digital)

Analog Input Signal Noise Elimination

- Averaging Method - Simple Technique
  - Sample Analog Input Multiple Times
  - Take the Average of the result
- Useful to eliminate the case of noise
  - Assuming analog input voltage does not change often
- Important: The average has to be made on readings that correspond to the same analog input voltage
  - You do NOT want to average values corresponding to different ADC levels

STM32 Oversampling

- STM32 Oversampling can be used for averaging
- Hardware performs a sum of a given number of ADC raw samples into one final sample
  - Hardware then right bit-shifts final value
  - Can process up to 1024 input samples

Three Ways To Read ADC

- Polling / Blocking Mode
  - Start conversion, then check when done
- Interrupts / Non-Blocking Mode
  - Start conversion, interrupt when done
  - Be careful of “interrupt flood” if you are taking lots of samples
- DMA / Non-Blocking Mode
  - Start conversion, DMA does transfer

Reference - Application Note

- https://www.st.com/resource/en/application_note/cd00211314-how-to-get-the-best-adc-accuracy-in-stm32-microcontrollers-stmicroelectronics.pdf

life.augmented

AN2834
Application note

How to get the best ADC accuracy in STM32 microcontrollers

Intro

- STM32 MCUs and ADC
  - 12-bit and 16-bit (depending on model)
  - Self-calibration
- ADC Accuracy can be hard with ADC
  - My project example: reading water level sensor in waste-water pipe and widely varying readings even when water level not changing
  - We want to know hardware and software techniques to minimize ADC errors

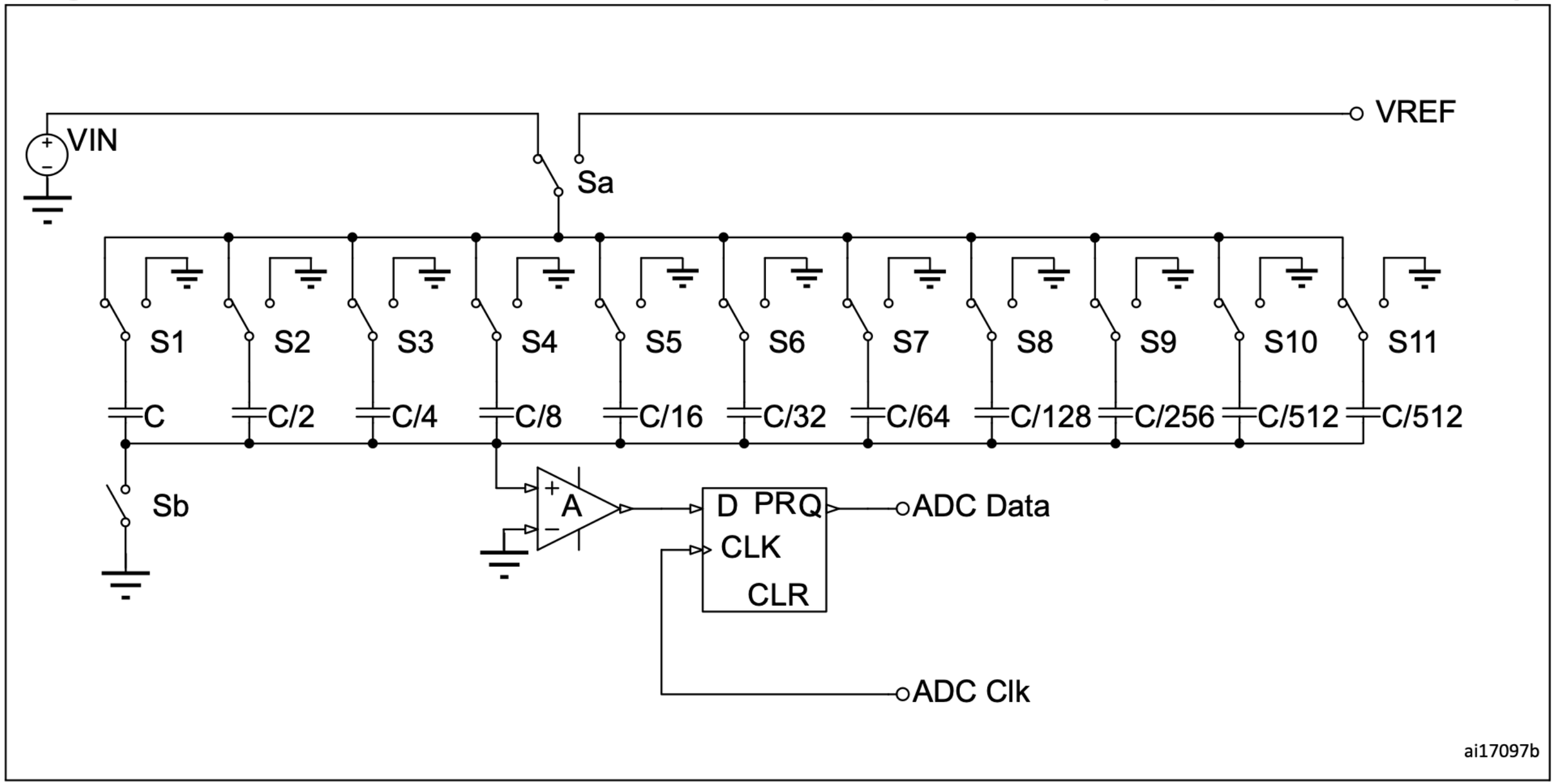
ADC Internal Operation

ADC Internal Operation

- STM32 ADC uses SAR
  - Successive Approximation Register
    - Conversion performed in several steps
      - Steps equal to number of bits in the ADC
        - Each Step driven by the ADC Clock
          - Each ADC clock produces 1 bit of output

10-bit SAR - Overview

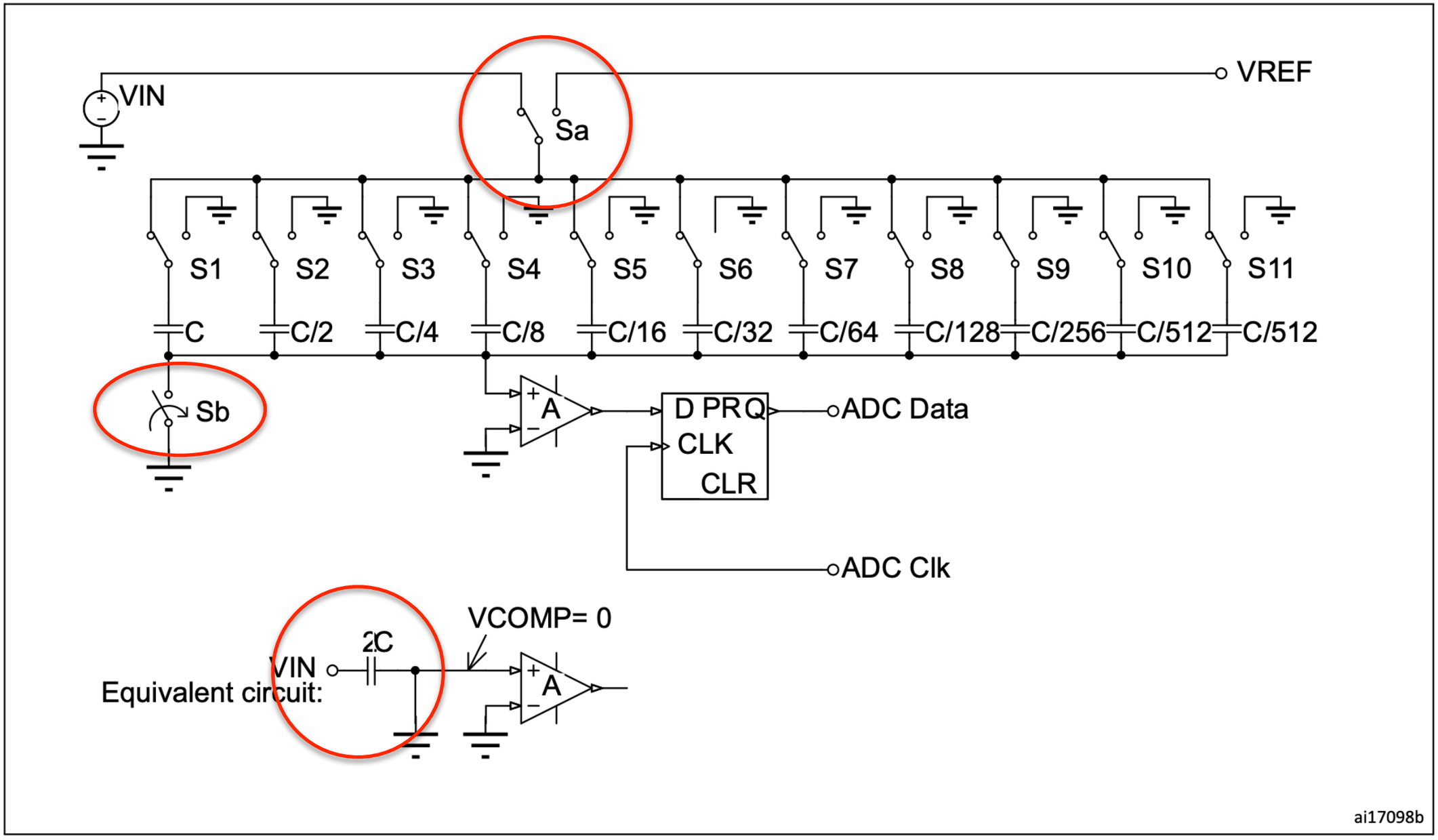
Figure 1. Basic schematic of SAR switched-capacitor ADC (example of 10-bit ADC)



1. Basic ADC schematic with digital output.

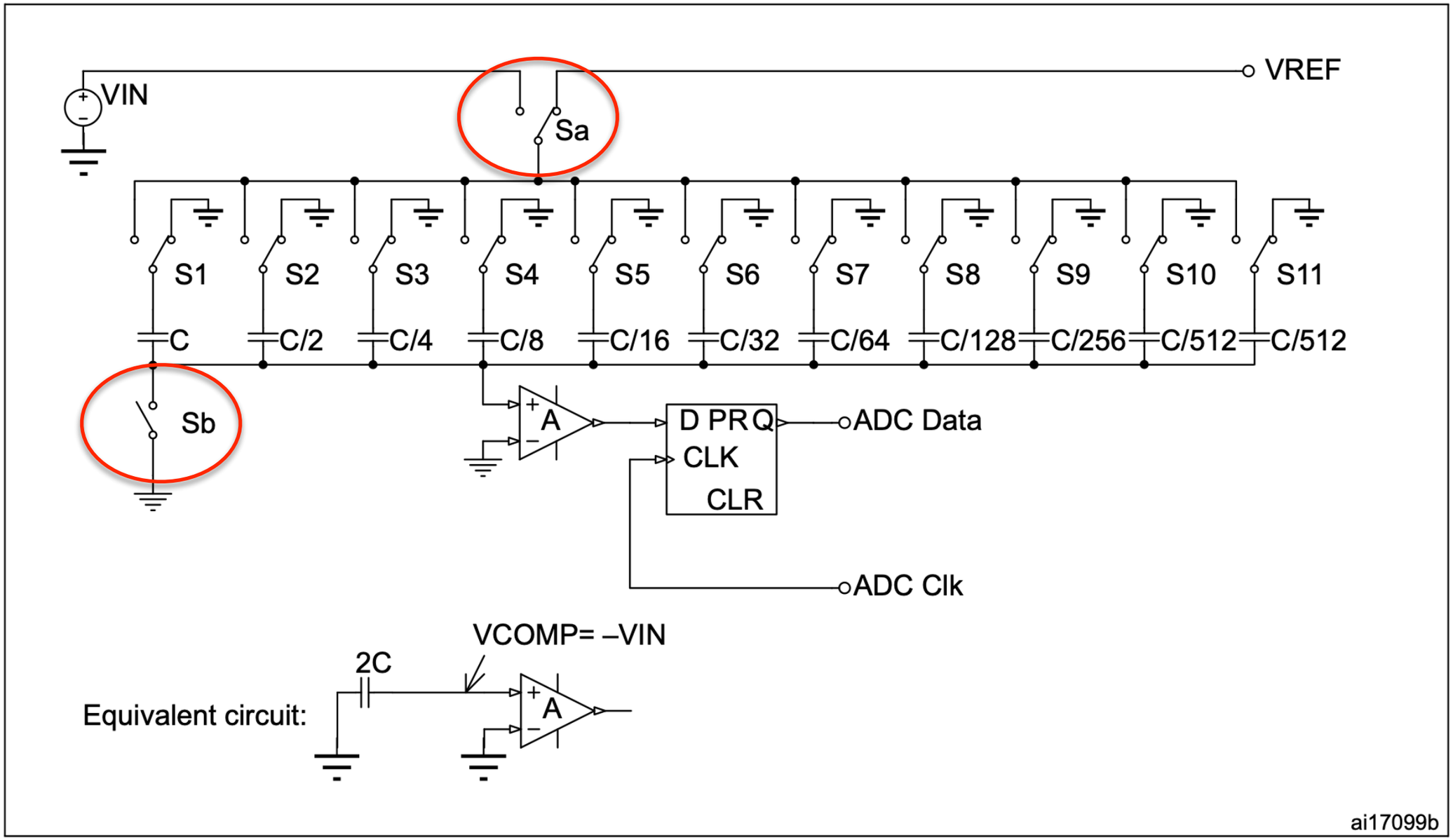
Sample State

Figure 2. Sample state



Hold State

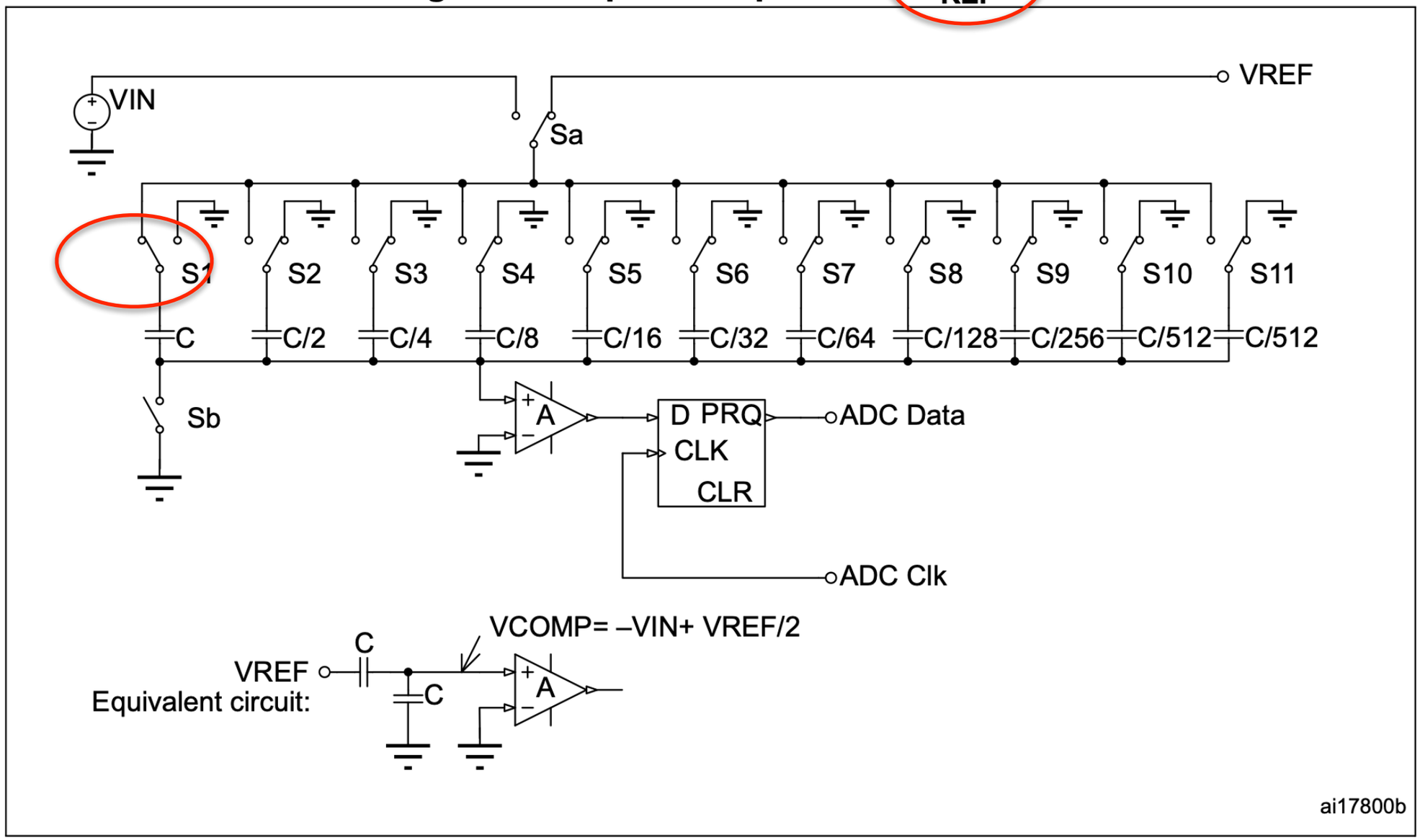
Figure 3. Hold state



1. Hold state: the input is disconnected, capacitors hold input voltage. Sb switch is open, then S1-S11

Step 1 Approximation

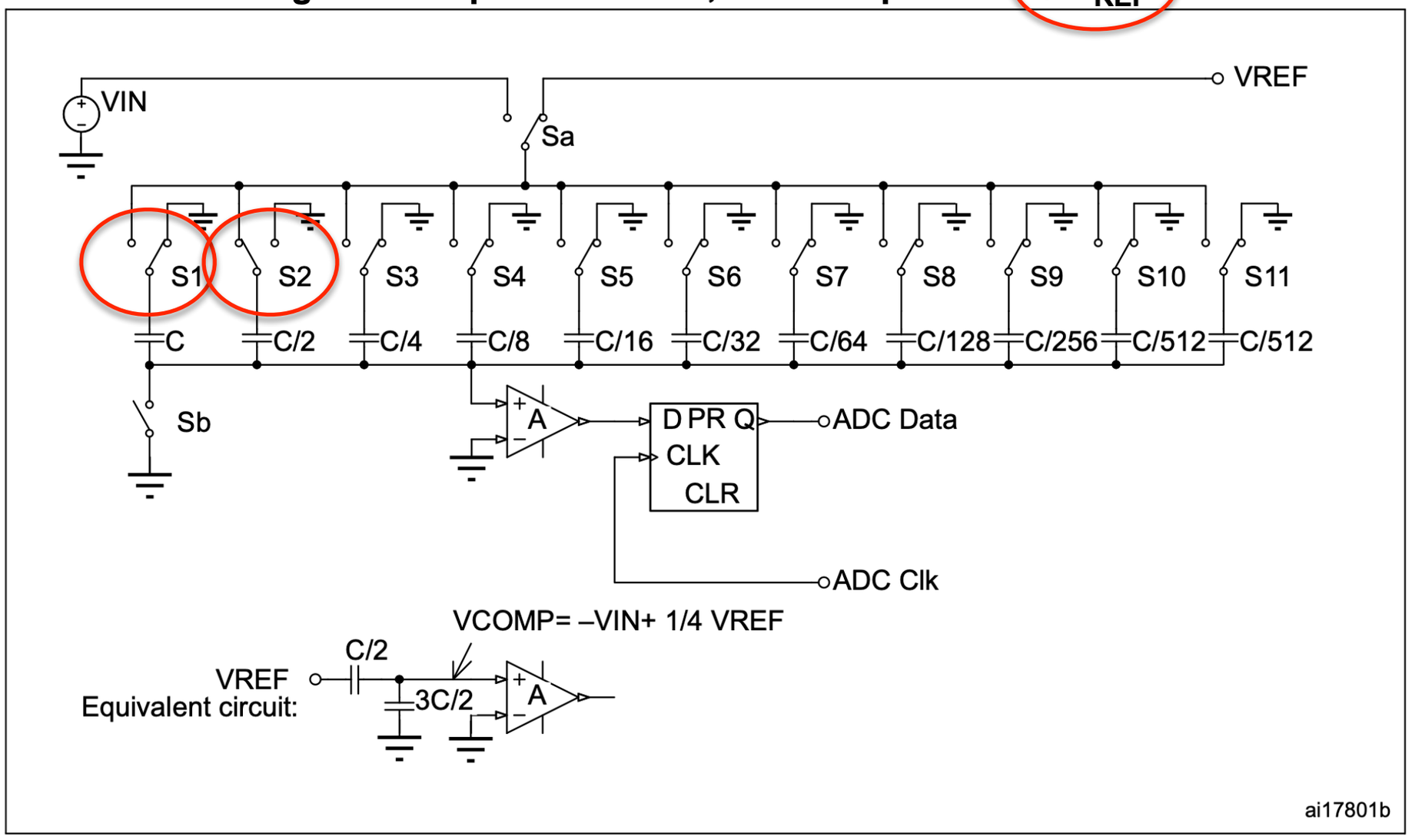
Figure 4. Step 1: Compare with $V_{REF}/2$



1. First approximation step. S1 switched to V_{REF}.

Step 2 (if MSB = 0) Approximation

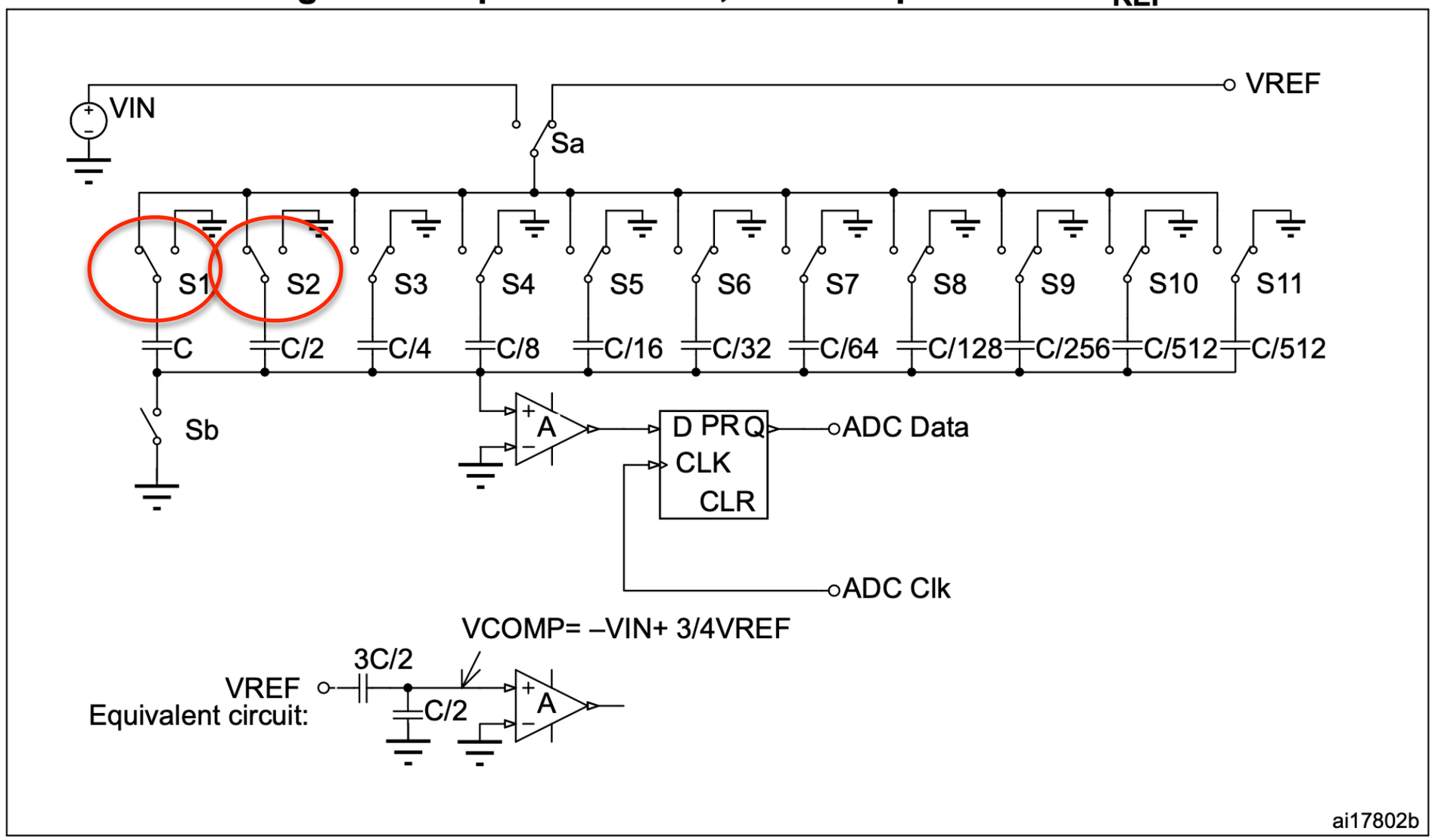
Figure 5. Step 2: If MSB = 0, then compare with ¼V_{REF}



1. Compare with ¼V_{REF}; if MSB =1. S1 switched back to ground. S2 switched to V_{REF}.

Step 2 (if MSB = 1) Approximation

Figure 6. Step 2: If MSB = 1, then compare with ¾V_{REF}



ADC Errors

ADC Errors

- Two Categories
  - Errors due to ADC
  - Errors due to Environment

Errors due to ADC

- Offset Error
- Gain Error
- Differential Linearity Error
- Integral Linearity Error
- Total Unadjusted Error

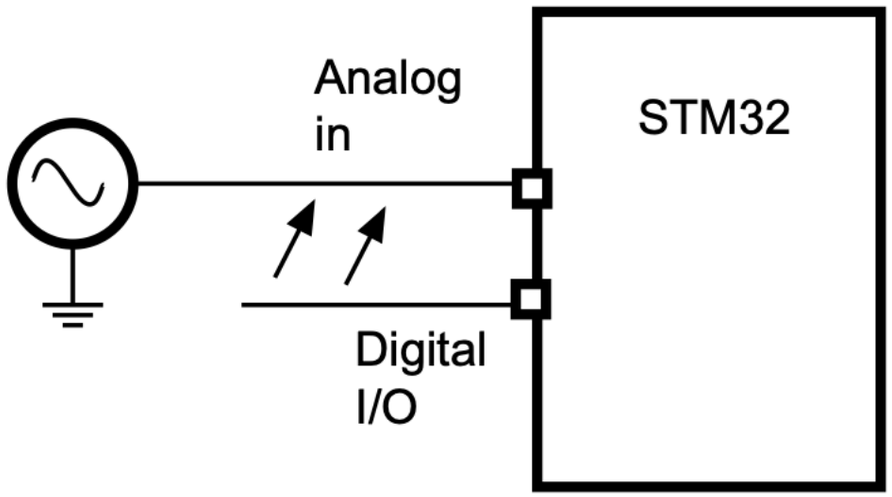
Errors due to ADC Environment

- Reference Voltage Noise
- Reference Voltage / Power Supply Regulation
- Reference Voltage Decoupling and Impedance
- External Reference Voltage Parameters
- Analog Input Signal Noise
- ADC dynamic range bad match for max input signal amplitude

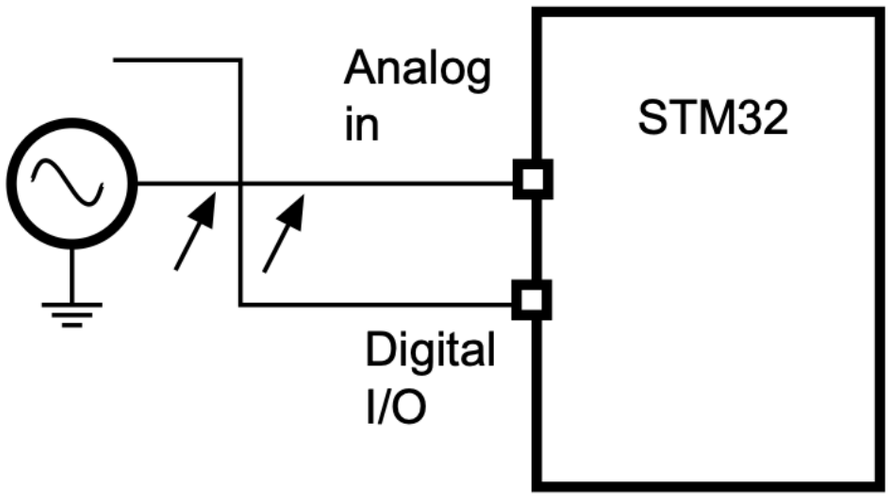
Temperature Influence

- Temperature has a major influence on ADC accuracy
  - Leads to two major errors
    - Offset Error drift
    - Gain Error Drift
- Compensate via firmware

I/O Pin Crosstalk



Case 1



Case 2

Much more in

life.augmented

AN2834
Application note

How to get the best ADC accuracy in STM32 microcontrollers