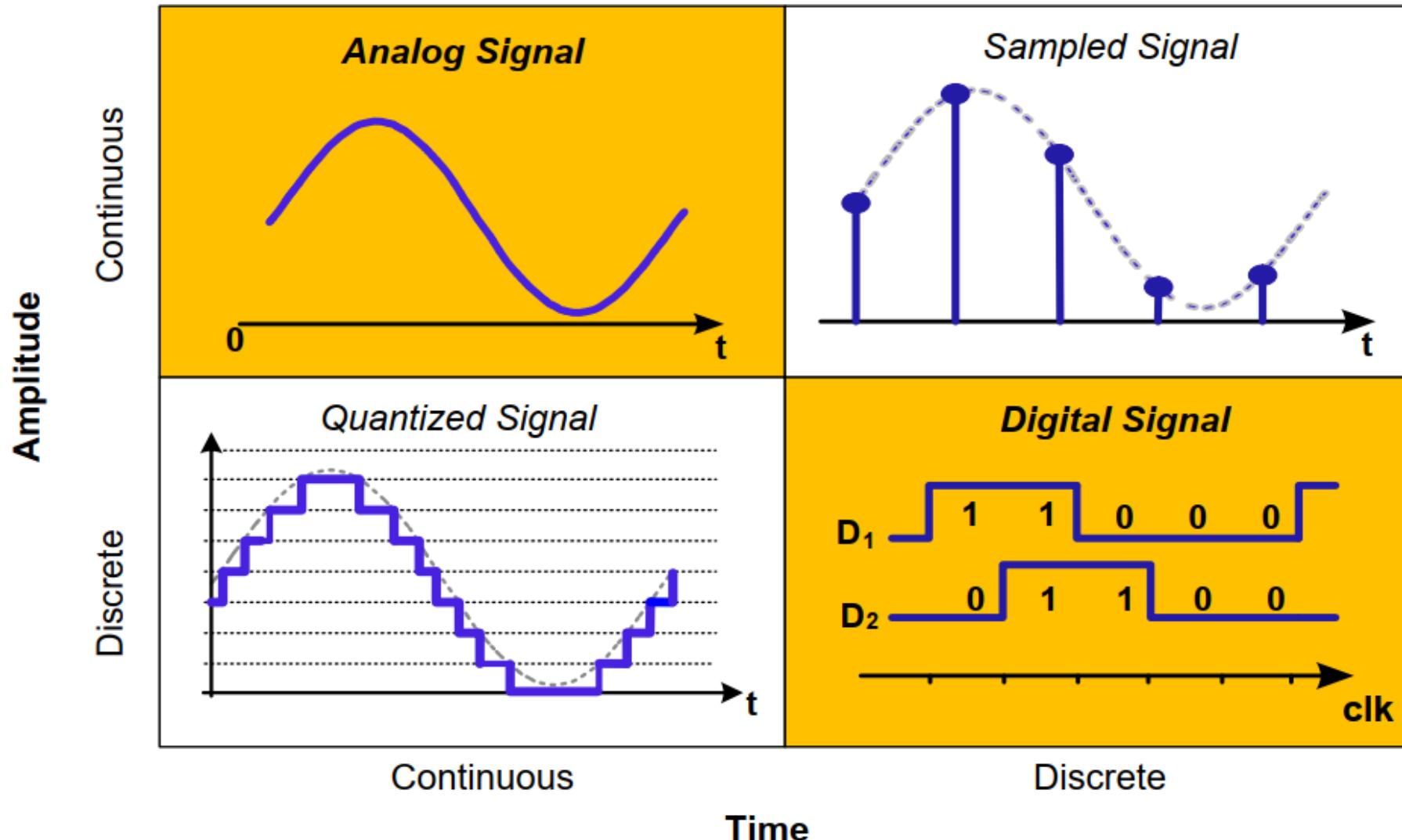


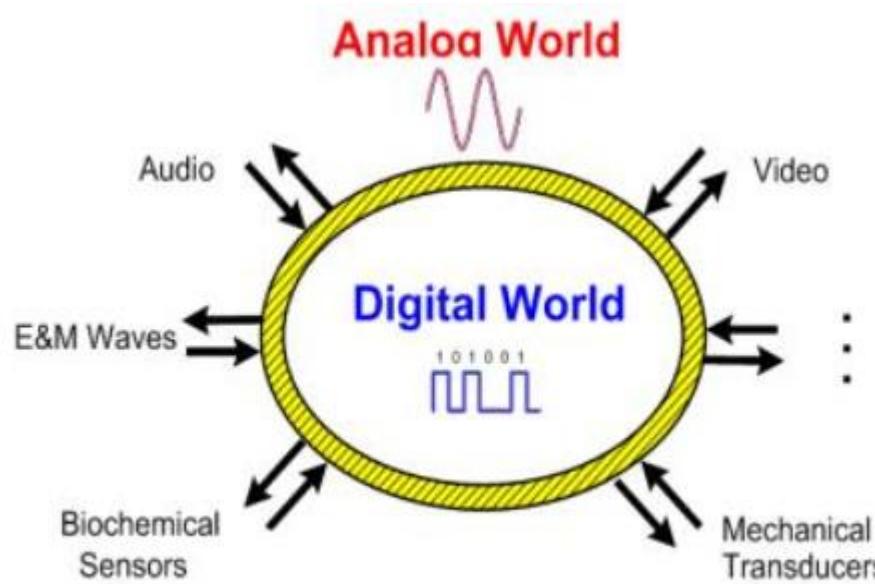
فصل پنجم : Analog to Digital Converter (ADC)

آنالوگ در مقایسه با دیجیتال

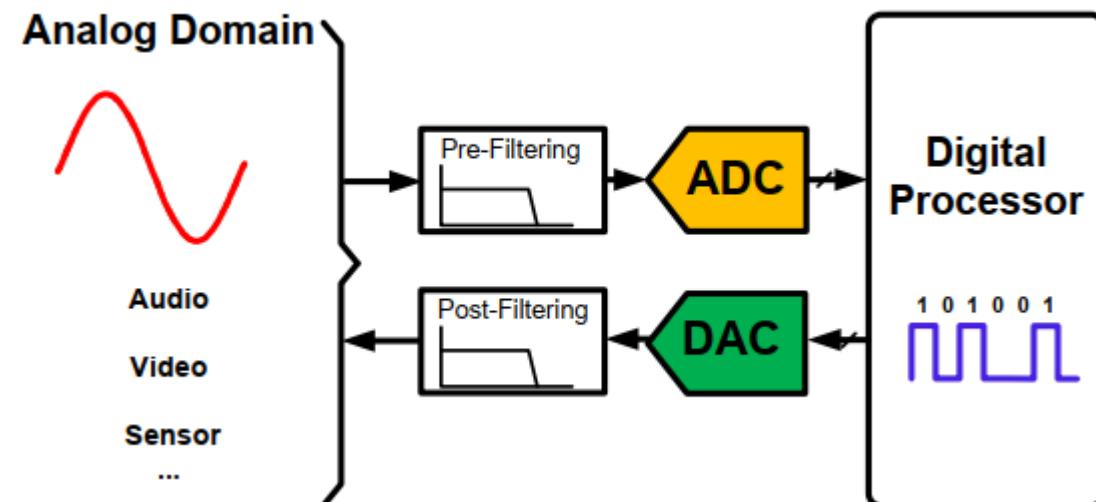


چرا مبدل‌های داده؟

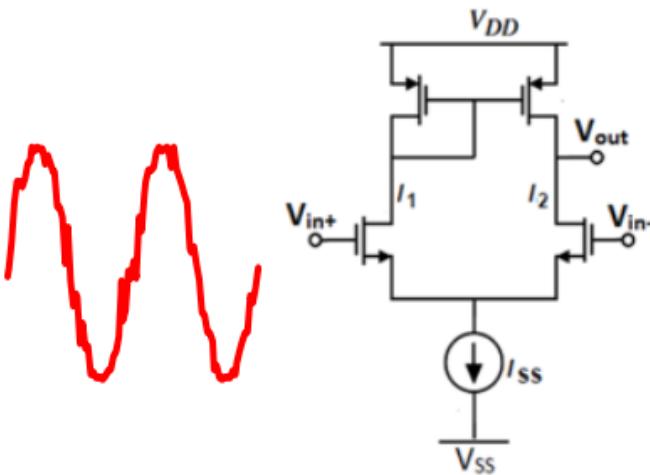
- پدیده‌های طبیعی همگی از نوع سیگنال‌های آنالوگ می‌باشند اما پردازنده‌ها فقط 0 یا 1 را می‌فهمند...
 - بنابراین نیاز به واحدی برای تبدیل سیگنال آنالوگ به دیجیتال (ADC) و بالعکس (DAC) وجود دارد!؟
 - سوال : چه نیازی است که سیگنال از آنالوگ به دیجیتال و سپس از دیجیتال به آنالوگ تبدیل شود!؟



Paul Gray's eggshell diagram



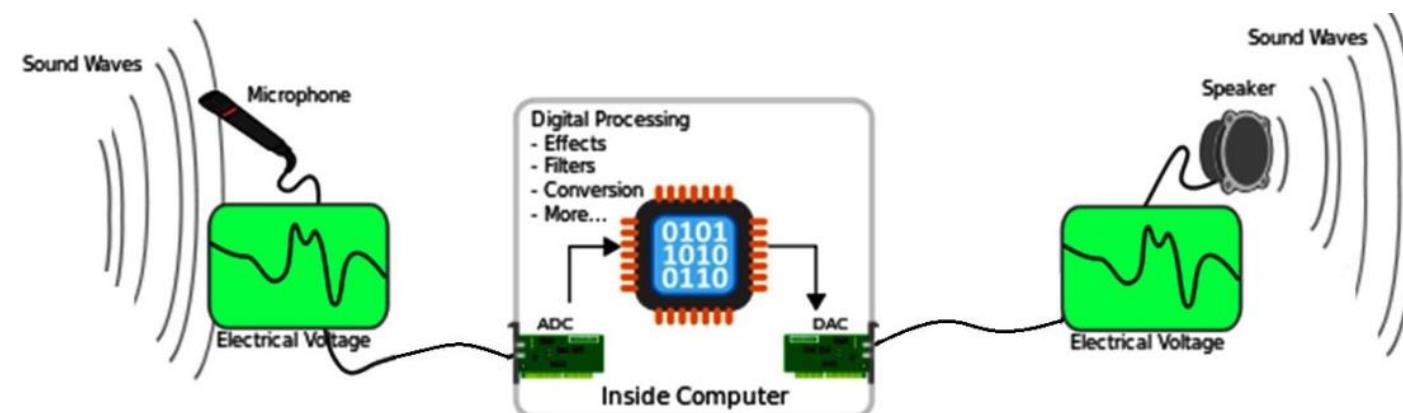
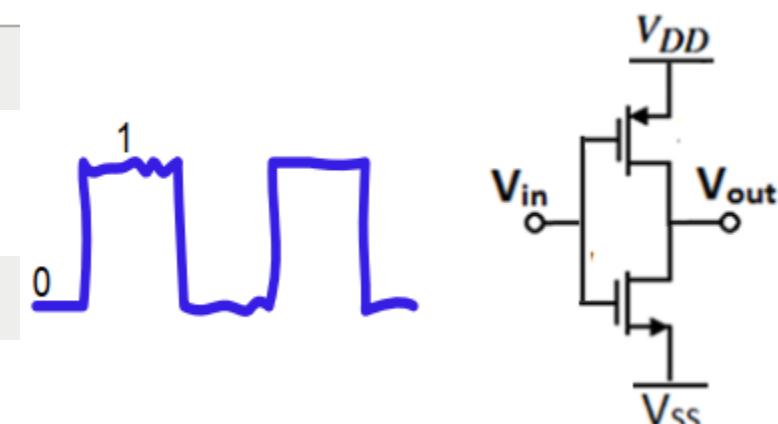
پردازش سیگنال‌های آنالوگ و دیجیتال



Analog signal processing

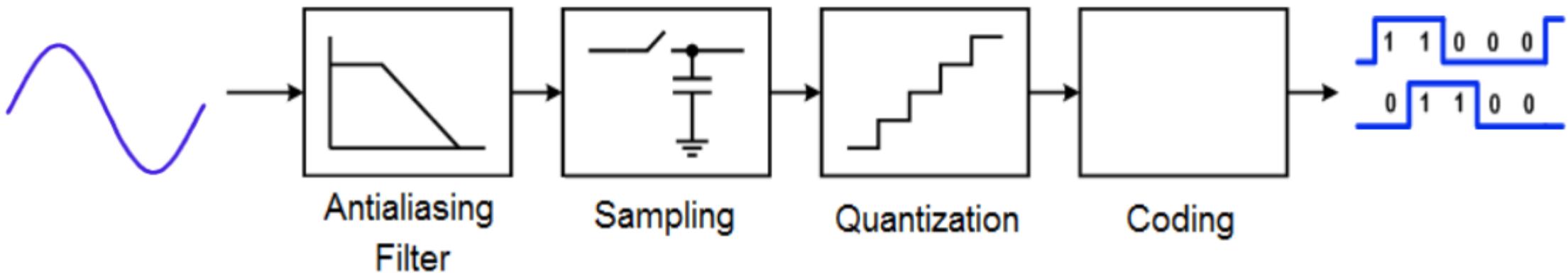
حساسیت به نویز بالا	حساسیت به نویز کم
حساس به اثرات غیرخطی و Mismatch	غیر حساس به اثرات غیرخطی و Mismatch
ذخیره سازی آسان و کم هزینه	ذخیره سازی دشوار و هزینه بر قابلیت طراحی و ساخت
اتوماتیک وجود ندارد	اتوماتیک را دارد

Digital signal processing

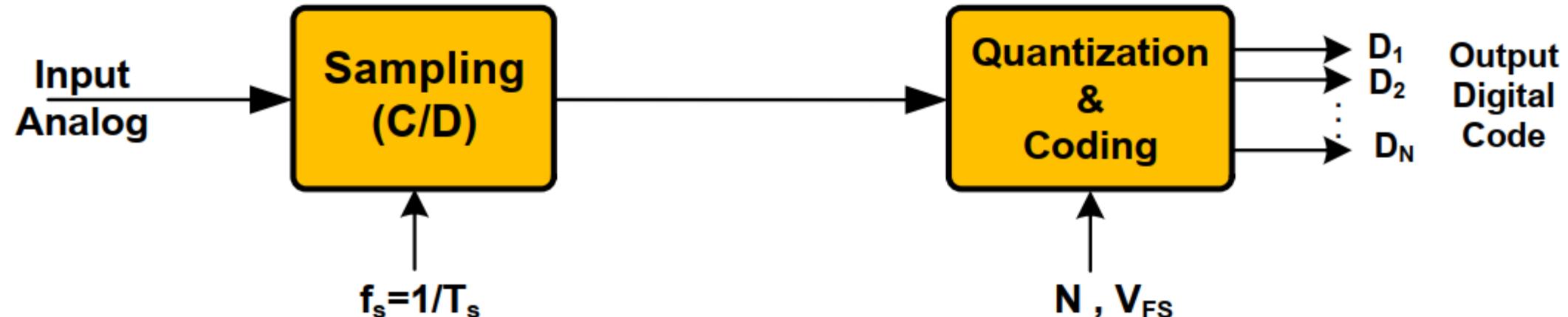
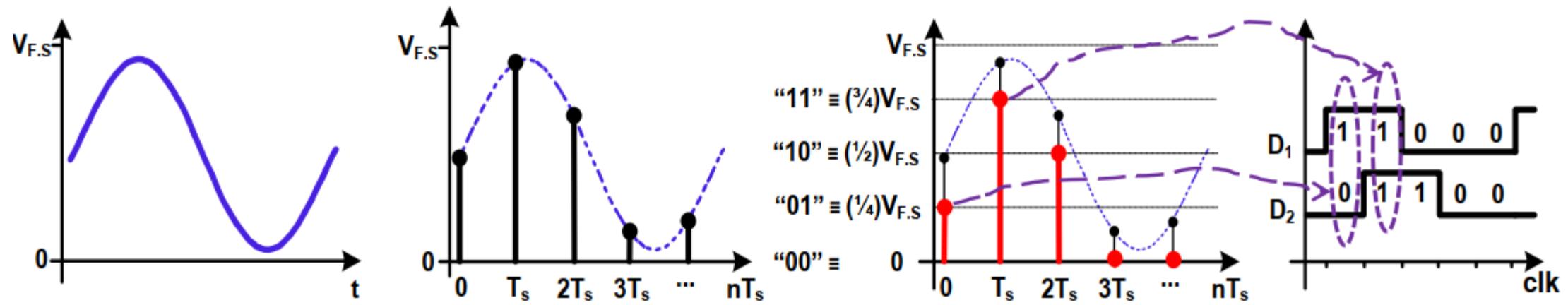


مراحل عملکرد مبدل آنالوگ به دیجیتال

- فیلتر : به منظور حذف فرکانس‌های اضافی
- نمونه برداری : ذخیره داده در لحظه مورد نظر
- تطبیق داده با سطوح نمونه‌برداری
- استخراج کد مناسب

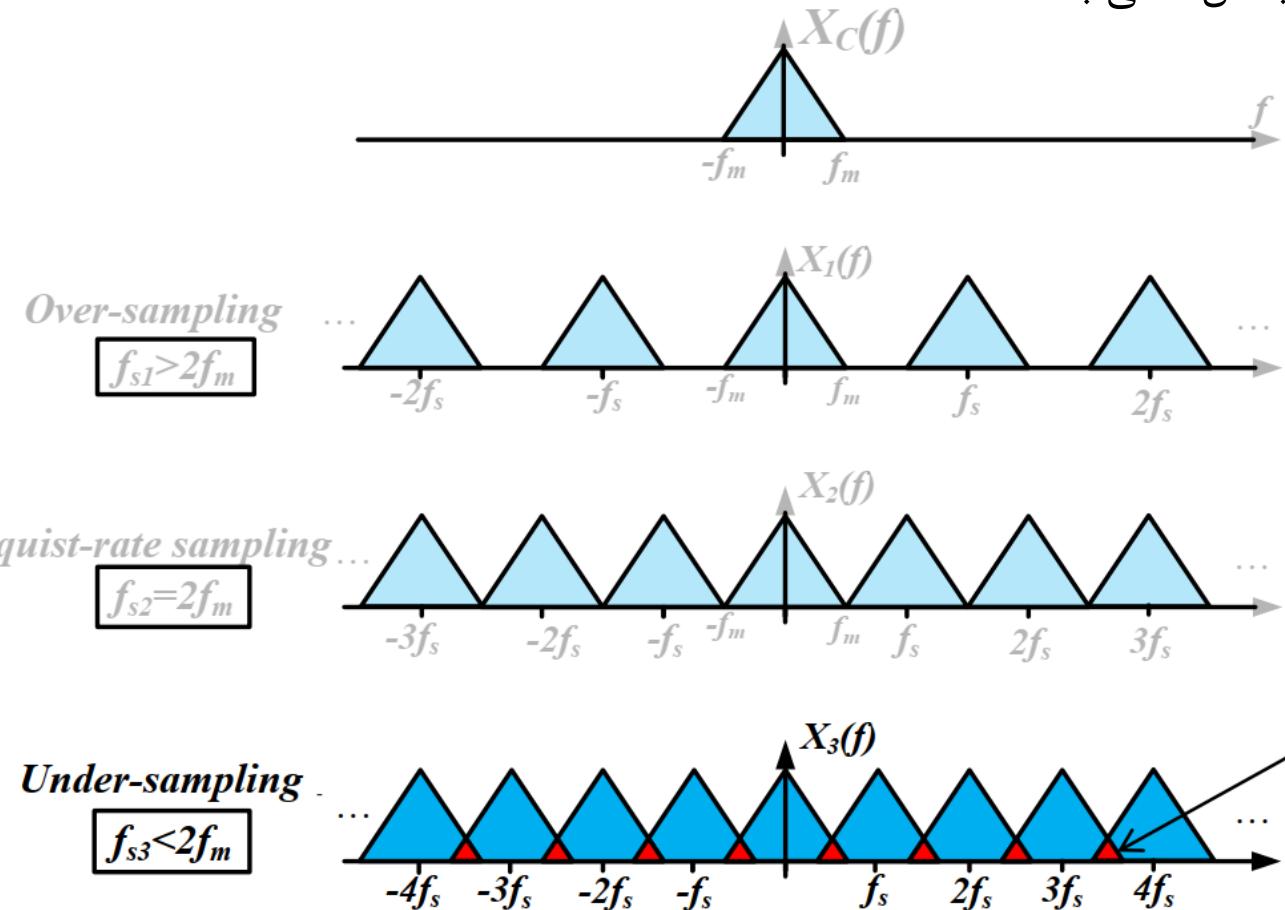
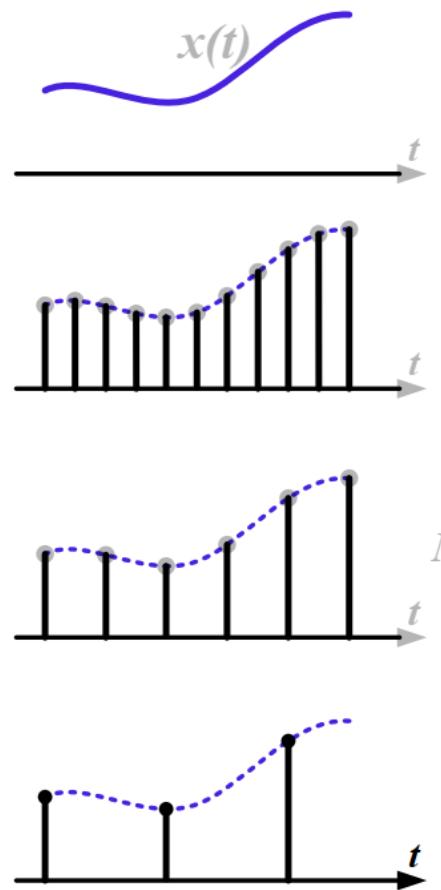


نمونه برداری و کوانتیزاسیون

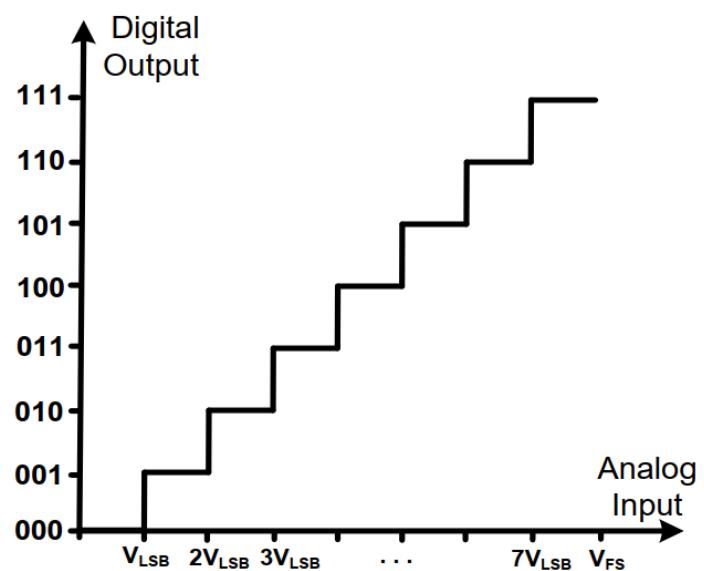
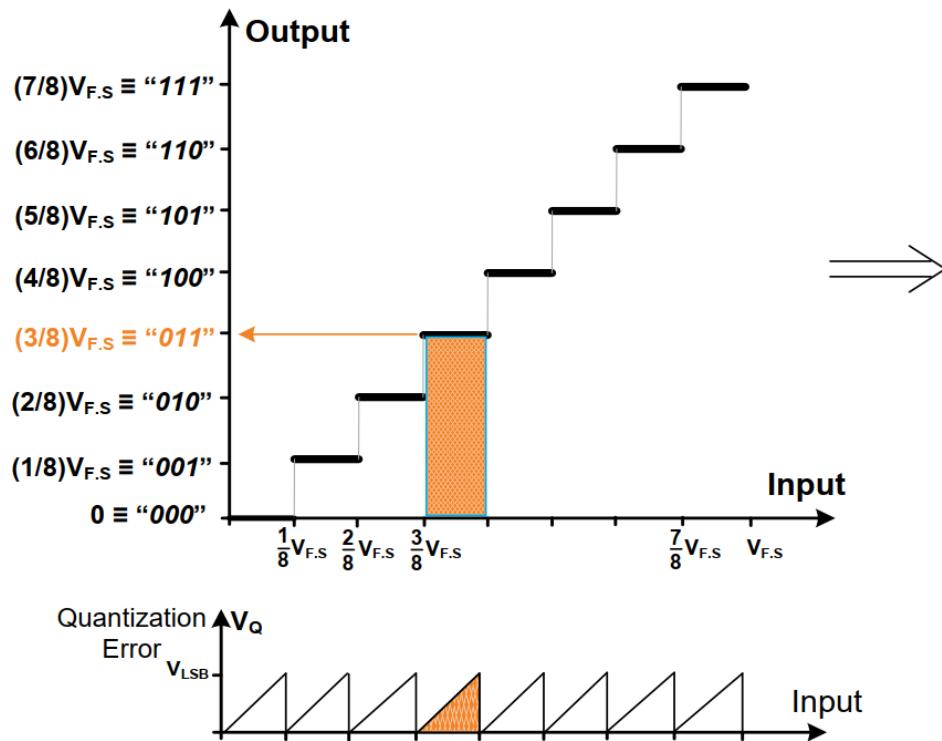


Nyquist نمونه برداری

- تئوری نمونه برداری نایکوئیست بیان می کند که برای نمونه برداری کامل از یک سگنال، فرکانس نمونه برداری، بایستی حداقل دو برابر فرکانس سیگنال اصلی باشد



کوانیزاسیون و استخراج کد



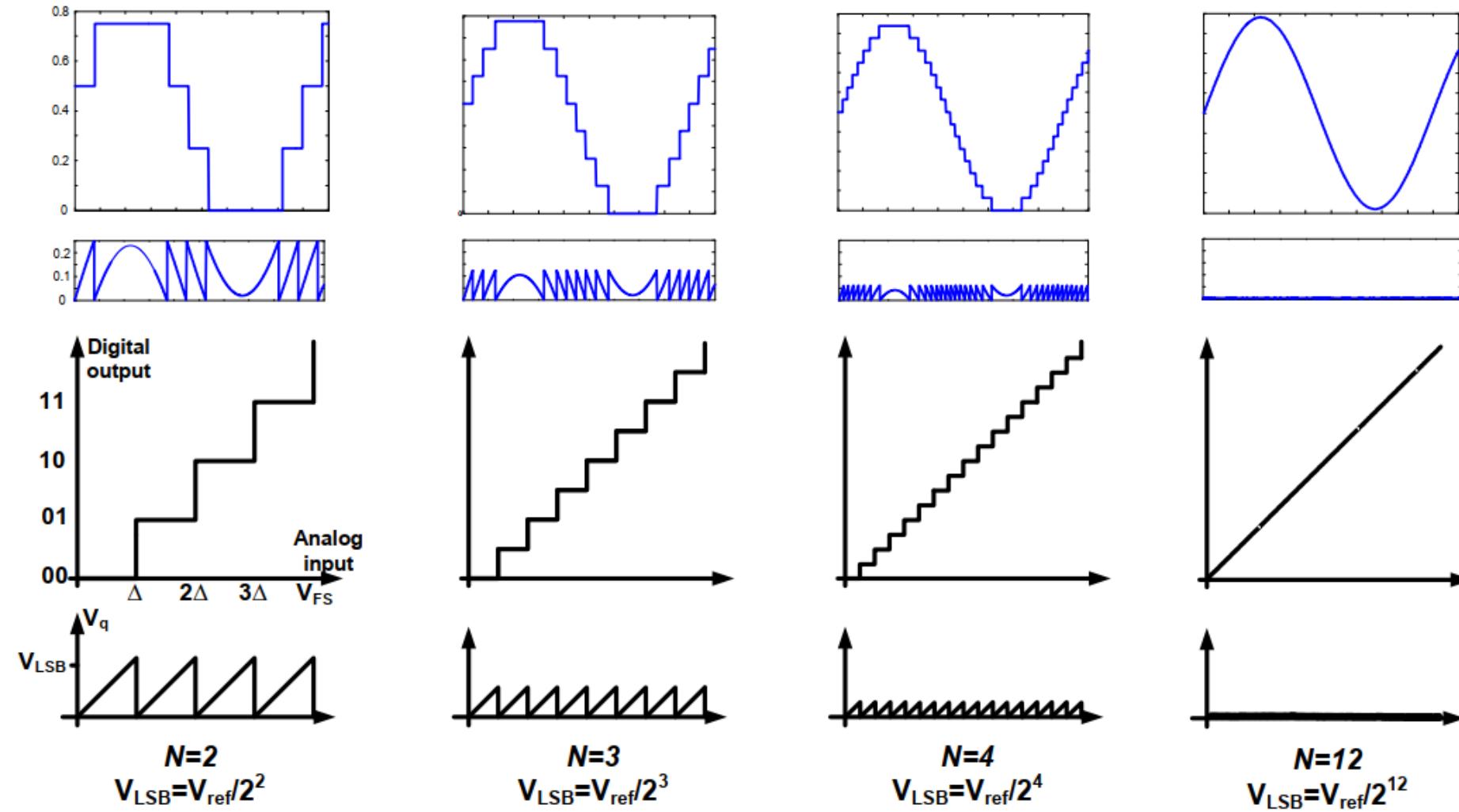
V(FS) : حداکثر ولتاژ سیگنال
ورودی قابل پذیرش

Resolution : دقت نمونه برداری
(مثال : نمونه بردار N بیت، V_{FS} را به 2^N سطح تقسیم می کند.)

Step-Size (V_{LSB}) : کوچکترین سطح ولتاژ قابل تغییر در ADC را به V_{LSB} می نامند.

Quantization Error : خطای بین مقدار اصلی سیگنال و مقدار نمونه برداری شده.

خطای کوانتیزاسیون



محاسبه SNR برای خطای کوانتیزاسیون

- با افزایش تعداد بیت‌های ADC، نویز کوانتیزاسیون کاهش می‌یابد. اما این کاهش تا کجا می‌تواند ادامه داشته باشد؟ باستی سیگنال از یک حدی باکیفیت‌تر باشد تا افزایش تعداد بیت نمونه‌برداری منطقی باشد...

$$SNR = SQNR = 20 \log \left(\frac{V_{sig,rms}}{V_{Q,rms}} \right) = 20 \log \left(\frac{\frac{A_{p-p}}{2\sqrt{2}}}{\frac{V_{FS}/2^N}{\sqrt{12}}} \right) = 6.02N + 1.76 + 20 \log \left(\frac{A_{p-p}}{V_{FS}} \right) \text{ dB}$$

N	SONR
8	50 dB
12	74 dB
16	98 dB
20	122 dB

if $A = V_{FS} \Rightarrow SNR_{max} = 6.02N + 1.76 \text{ dB}$

محاسبه SNR برای خطای کوانتیزاسیون

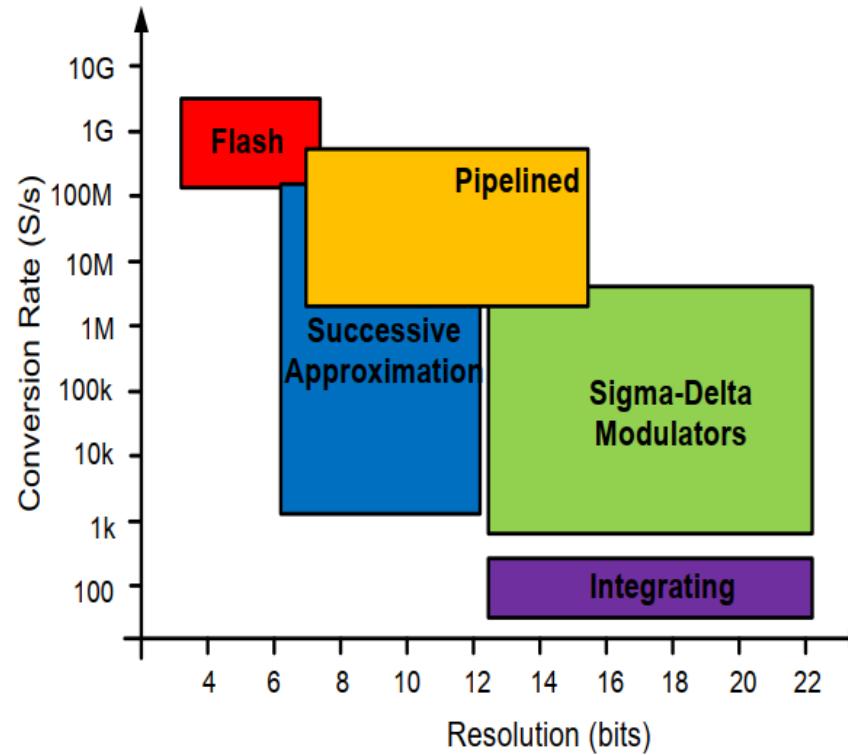
- برای یک ADC با رزولوشن N بیت و شرایط ایدهآل، مقادیر VLSB را محاسبه کنید! (با شرط $V_{FS} = 1V$)

N	8	10	12	14	16	19
V_{LSB}	3.92mV	977uV	244uV	61uV	15uV	1.9uV

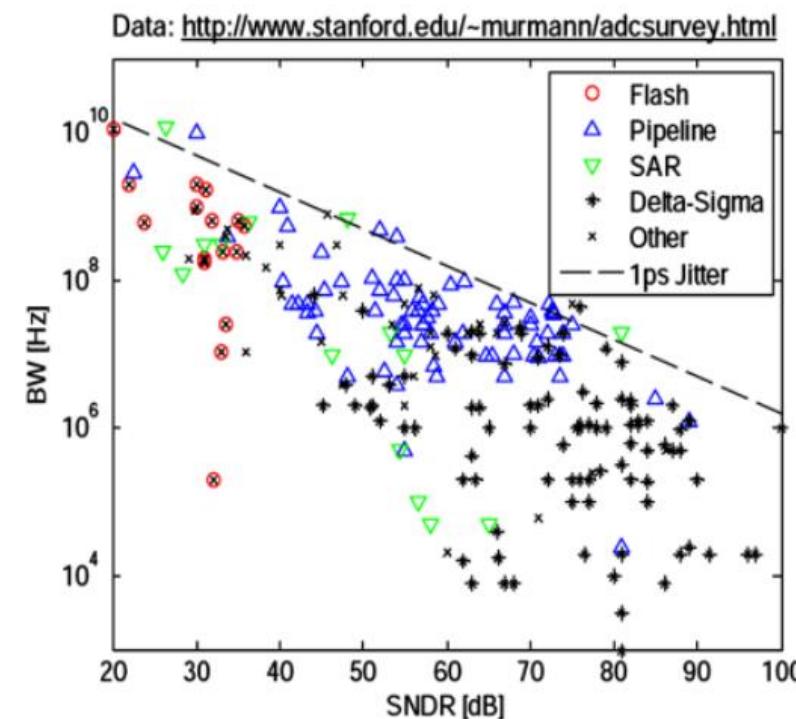
$$V = \frac{Q}{C} = \frac{nq_e}{C} = 12 \cdot \frac{1.602 \times 10^{-19} [\text{C}]}{1 \times 10^{-12} [\text{F}]} = 1.9 \mu\text{V}$$

A few electrons make big difference!!!

ADC انواع



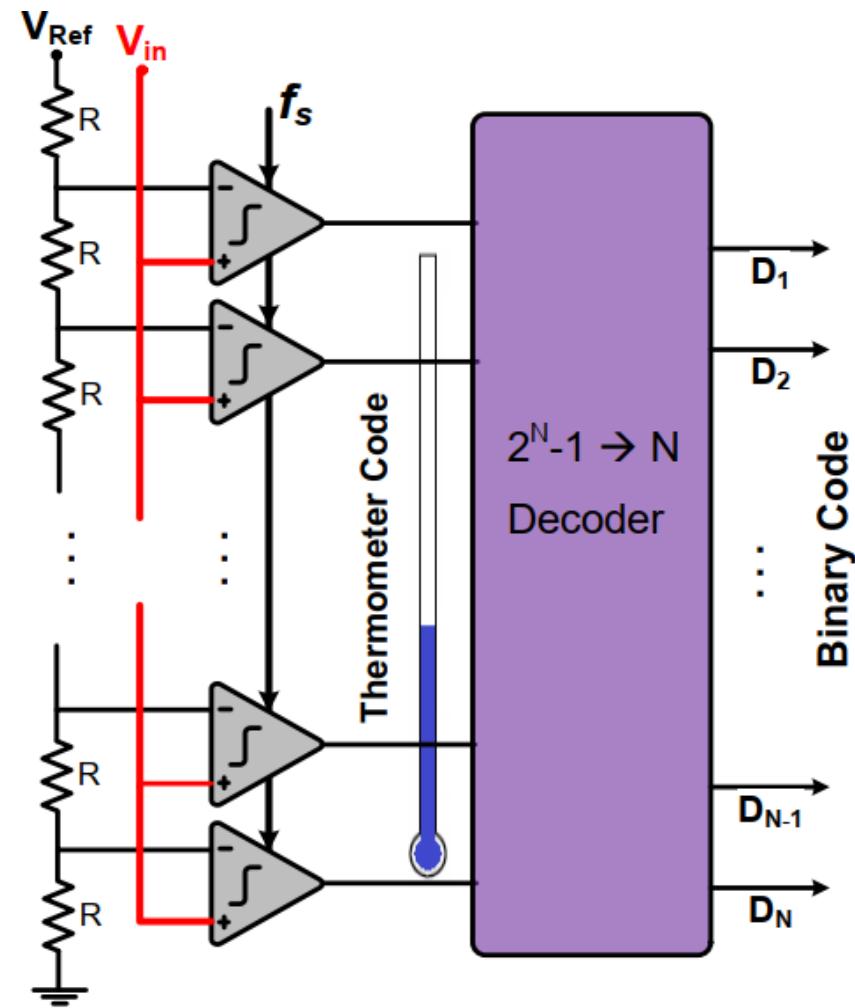
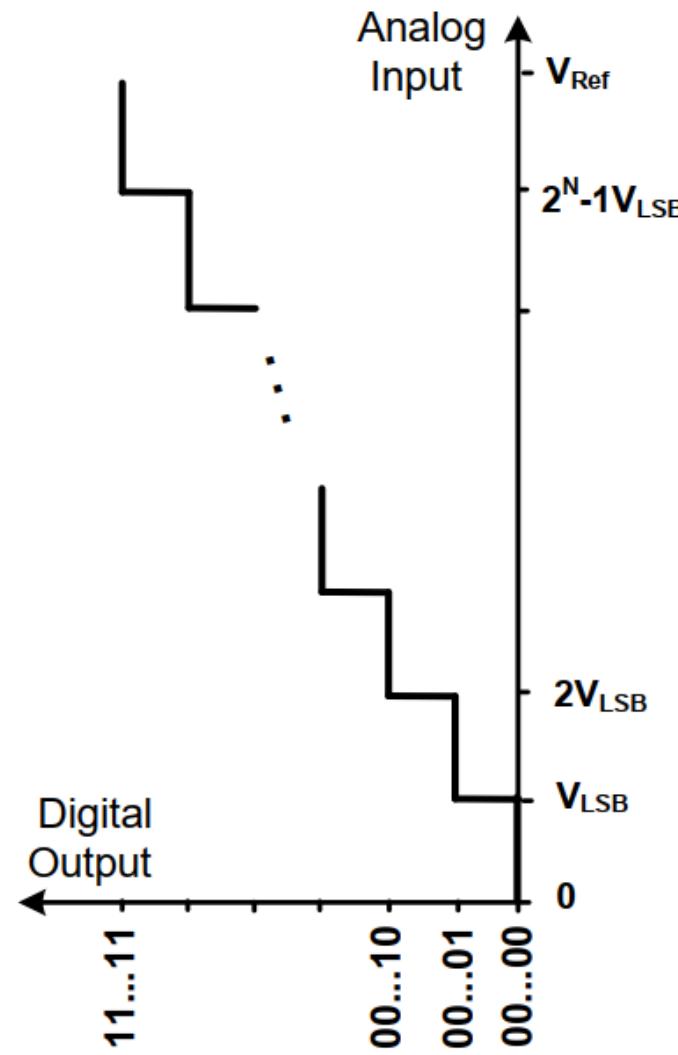
بر اساس تopolوژی و روش پیاده‌سازی، ADC‌ها به چندین نوع طبقه‌بندی می‌شوند.
: ($f_s \geq f_m$) ADC‌های با نرخ نایکوئیست



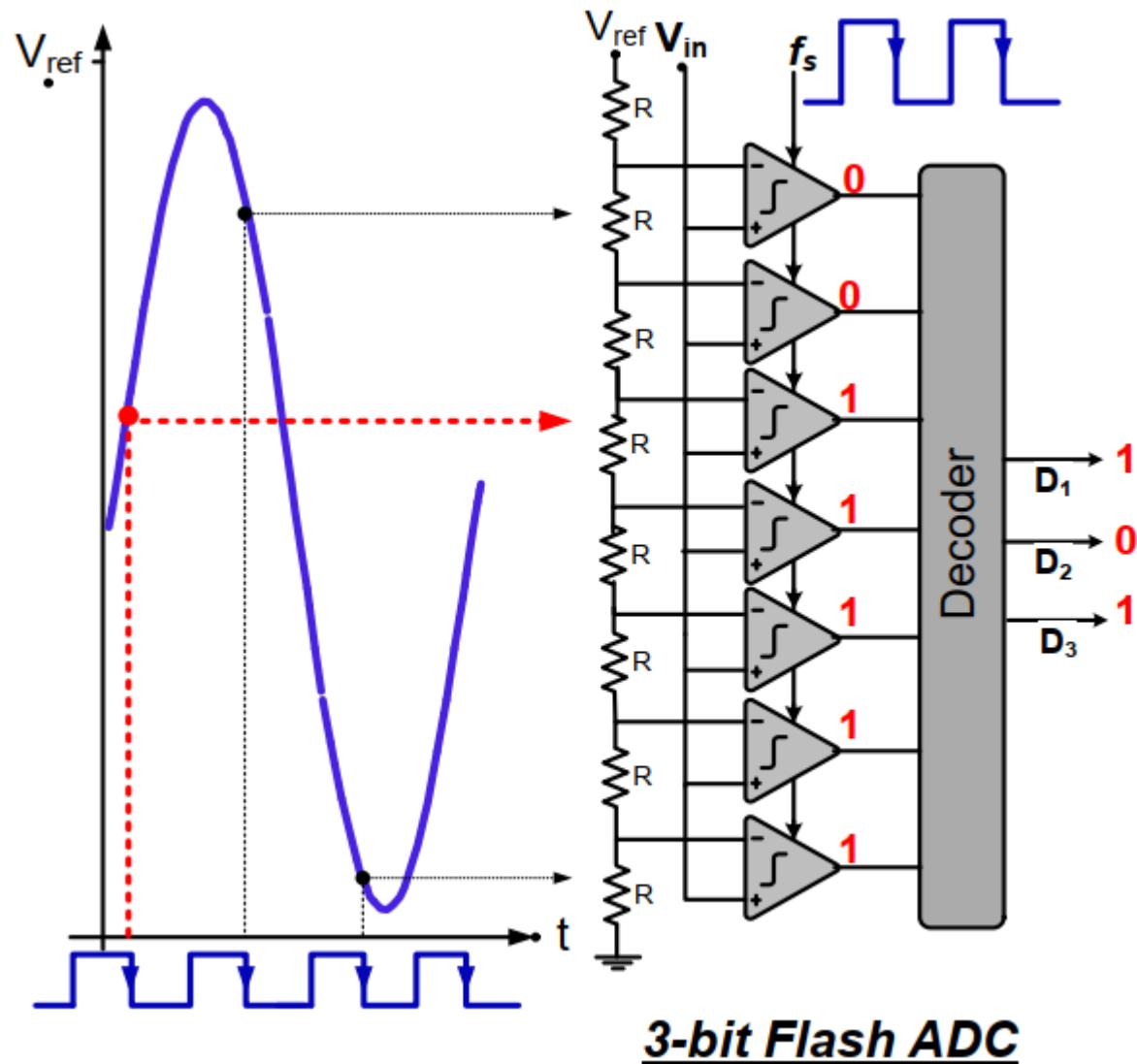
ISSCC & VLSI 1997-2008

- بر اساس توپولوژی و روش پیاده‌سازی، ADC‌ها به چندین نوع طبقه‌بندی می‌شوند.
: ($f_s \geq f_m$) ADC‌های با نرخ نایکوئیست :
- FLASH ADC
- Pipeline ADC
- SAR ADC
- Integrated Time ADC
- با نرخ بالاتر از نایکوئیست :
- Sigma-Delta ADC

Flash ADC عملکرد



عملکرد Flash ADC



- ورودی با $2^N - 1$ سطح ولتاژ مقایسه می‌شود.

- سرعت بالا

- Latency = $T_s = 1/f_s$

- Throughput = f_s

- رزولوشن پایین تا متوسط (تا 6 بیت)

- پیچیدگی بالا ($2^N - 1$ مقایسه کننده)

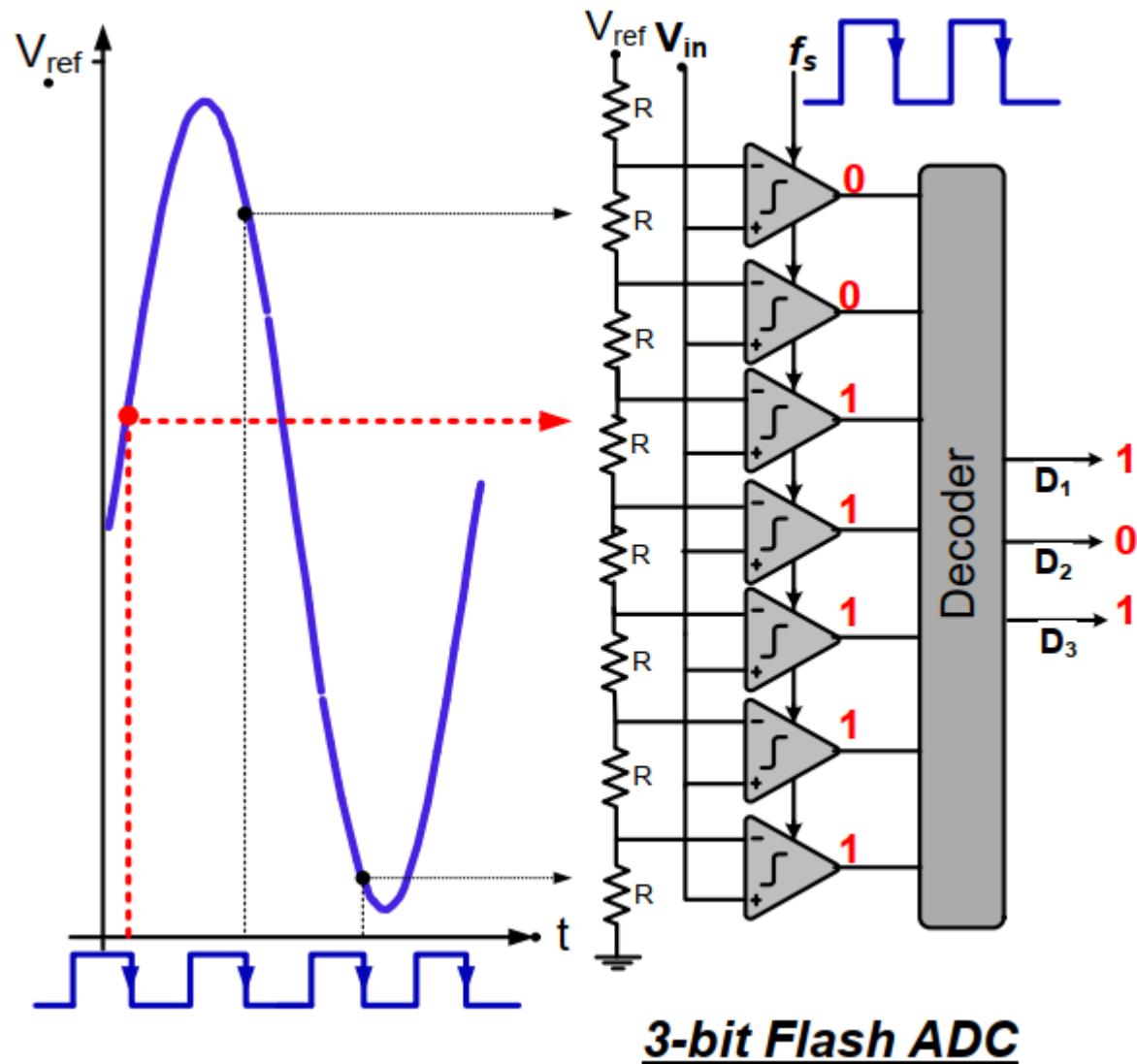
- مشکلات پیاده‌سازی :

- Mismatch

- اندازه مقاومت‌ها

- Jitter

عملکرد Flash ADC



- ورودی با $2^N - 1$ سطح ولتاژ مقایسه می‌شود.

- سرعت بالا

- Latency = $T_s = 1/f_s$

- Throughput = f_s

- رزولوشن پایین تا متوسط (تا 6 بیت)

- پیچیدگی بالا ($2^N - 1$ مقایسه کننده)

- مشکلات پیاده‌سازی :

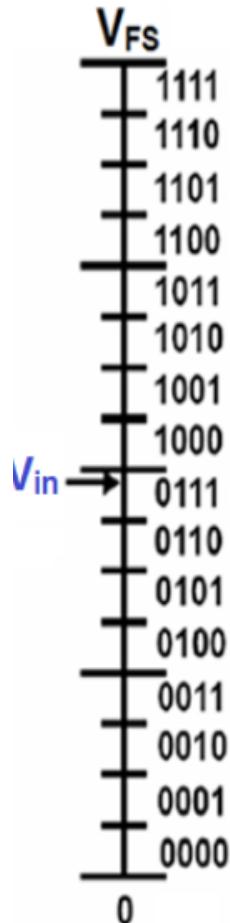
- Mismatch

- اندازه مقاومت‌ها

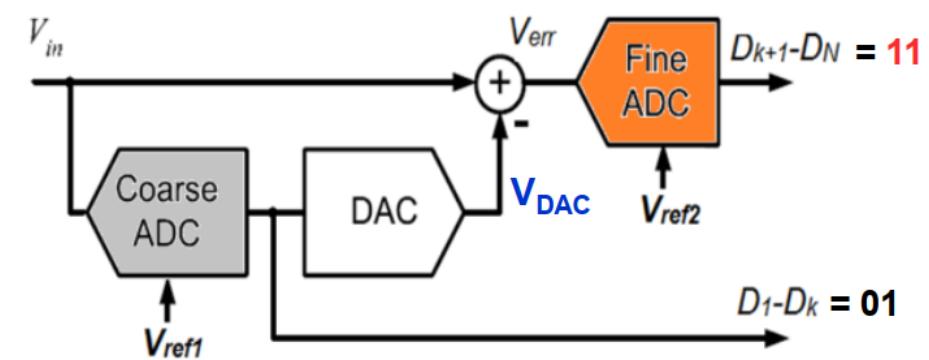
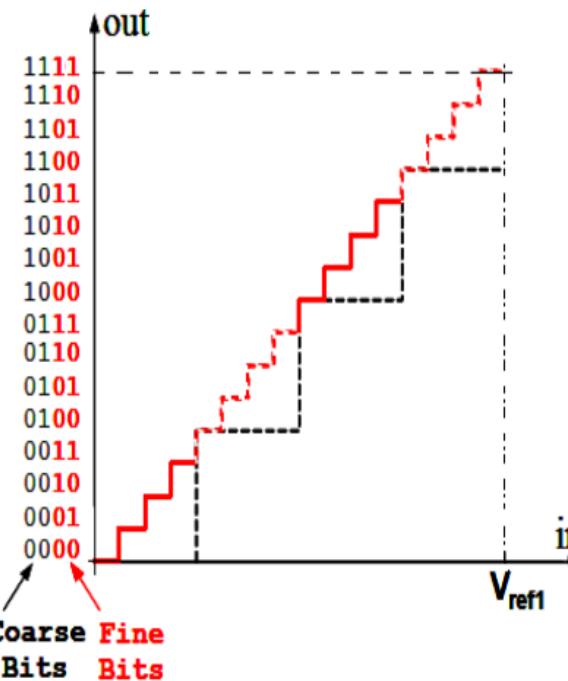
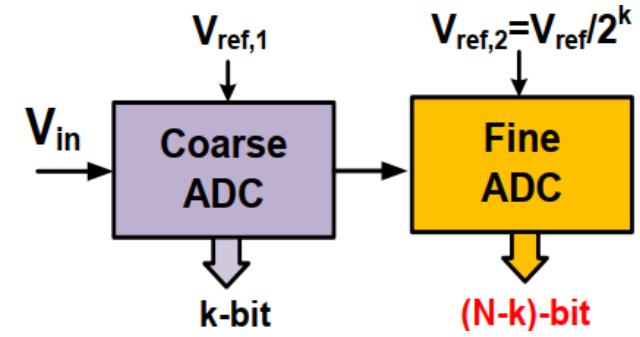
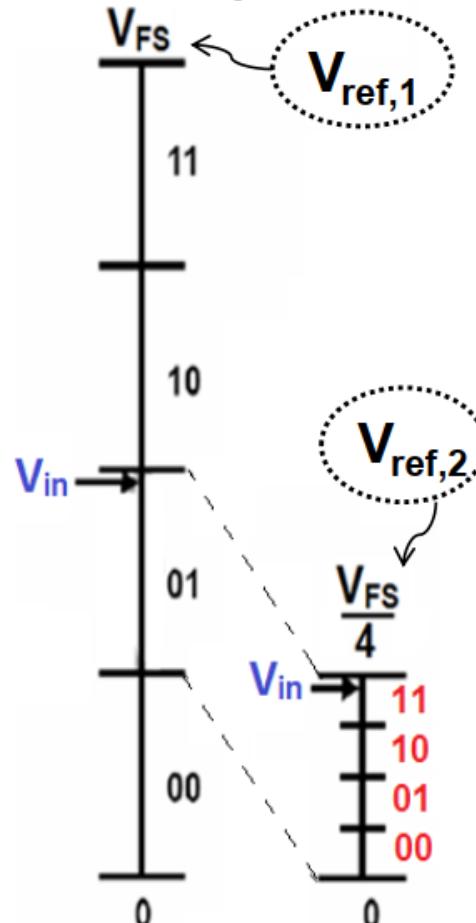
- Jitter

تئوری Pipeline ADC

**4-bit
flash ADC**



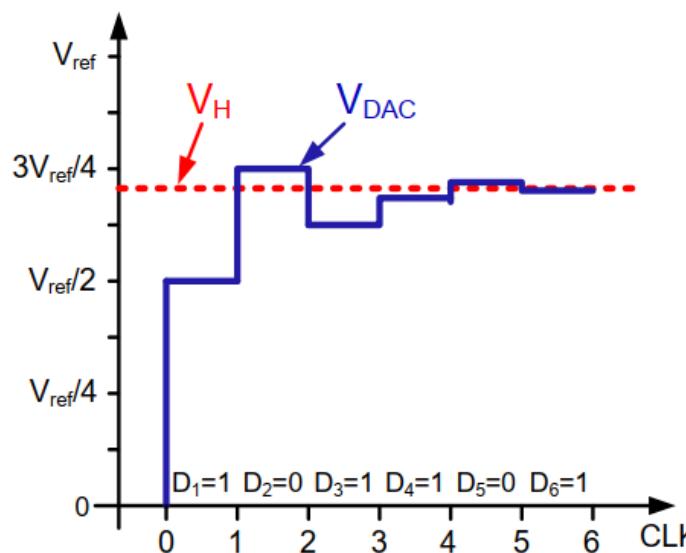
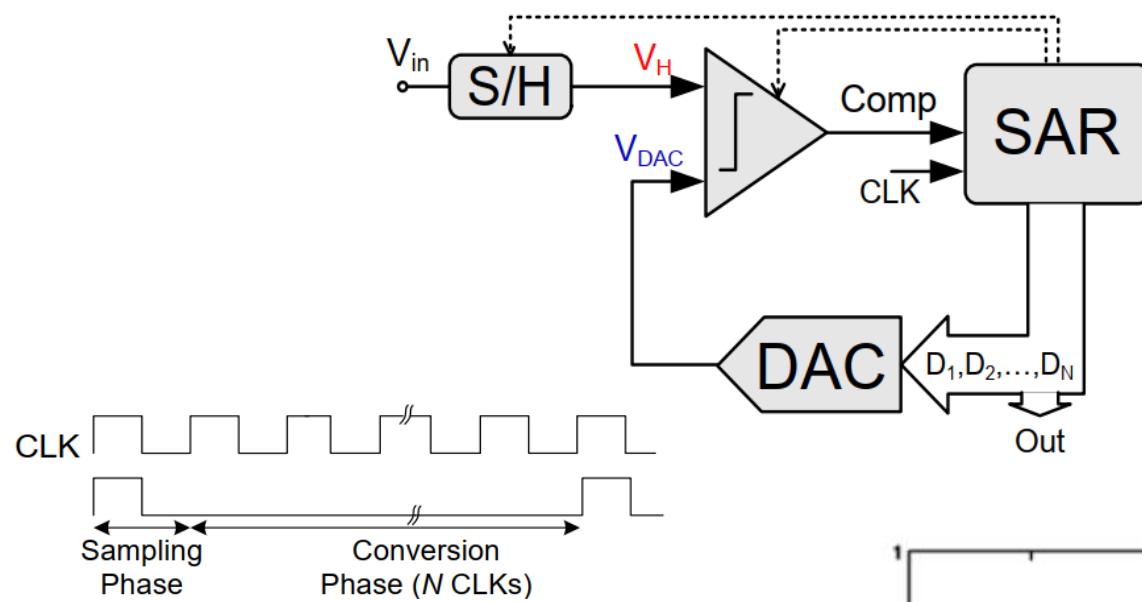
**4-bit
two-step ADC**



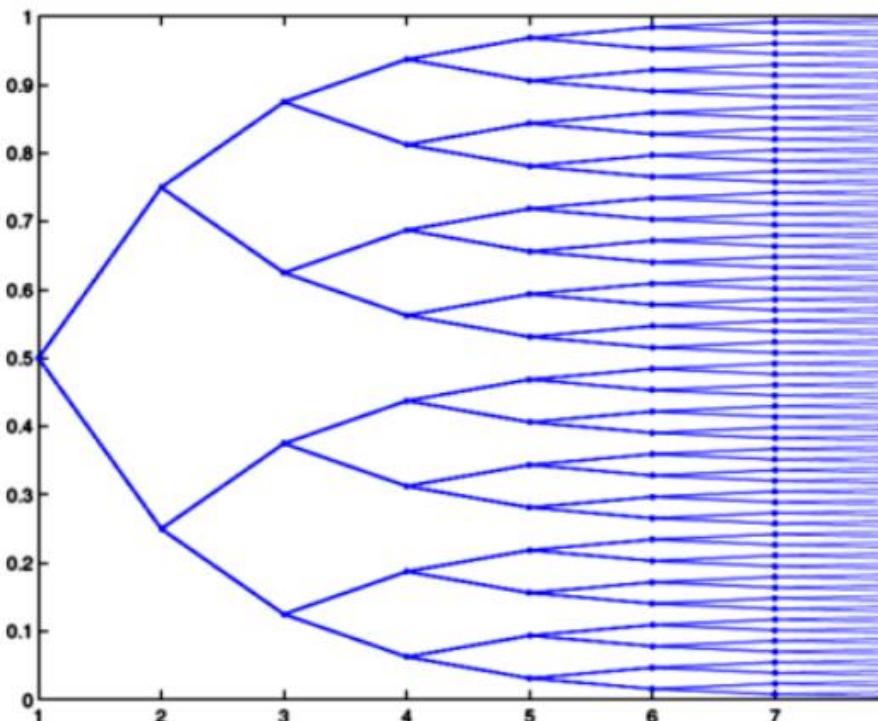
- تقسیم عملکرد بین سخت افزارها

- محاسبه بیت ها به صورت مجزا (مشکل!?)

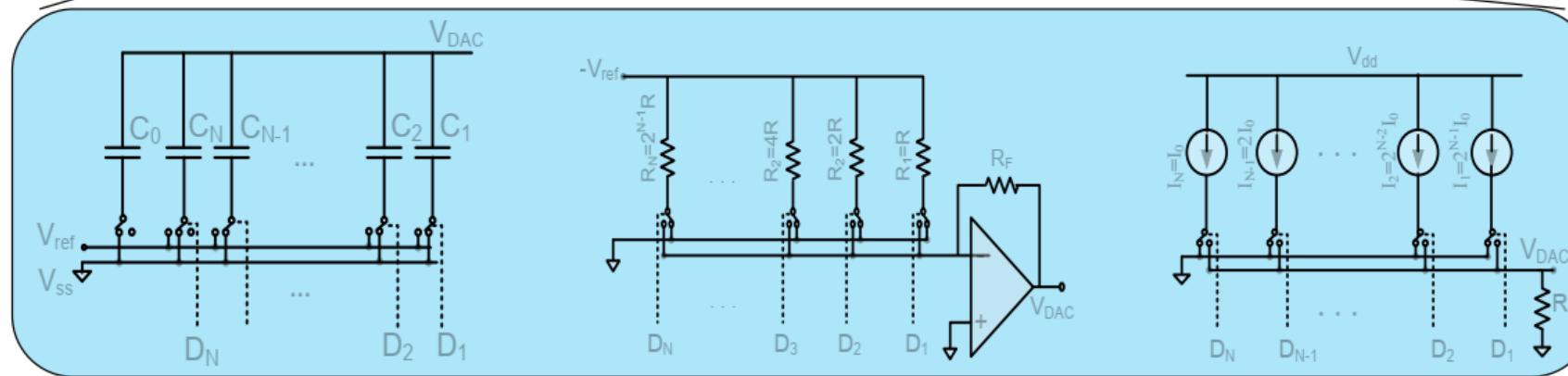
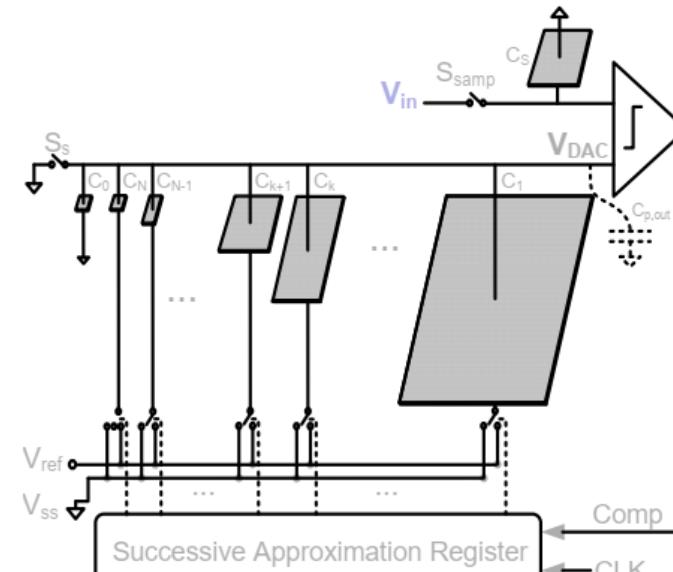
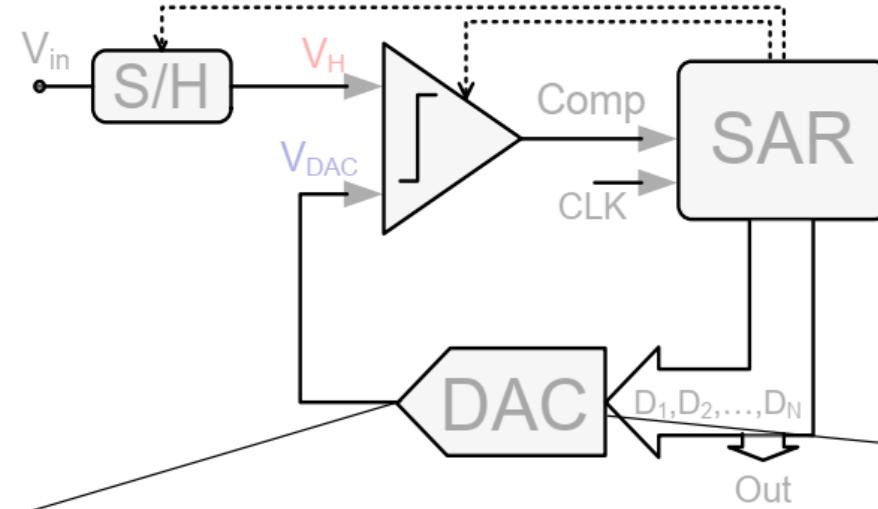
SAR ADC تئوری



- نحوه عملکرد SAR ADC‌ها مبتنی بر تخمین متوالی داده‌ها می‌باشد. به اینصورت که در هر مرحله، سیگنال ورودی با مقدار نصف ولتاژ رفرنس مقایسه می‌گردد. در صورت بزرگتر بودن، بیت 1 و در صورت کوچکتر بودن بیت 0 تولید می‌شود.



SAR ADC تئوری



- رایج‌ترین DAC بکاررفته در SAR ADC ها DAC خازنی می‌باشد.
- مزایا : ?

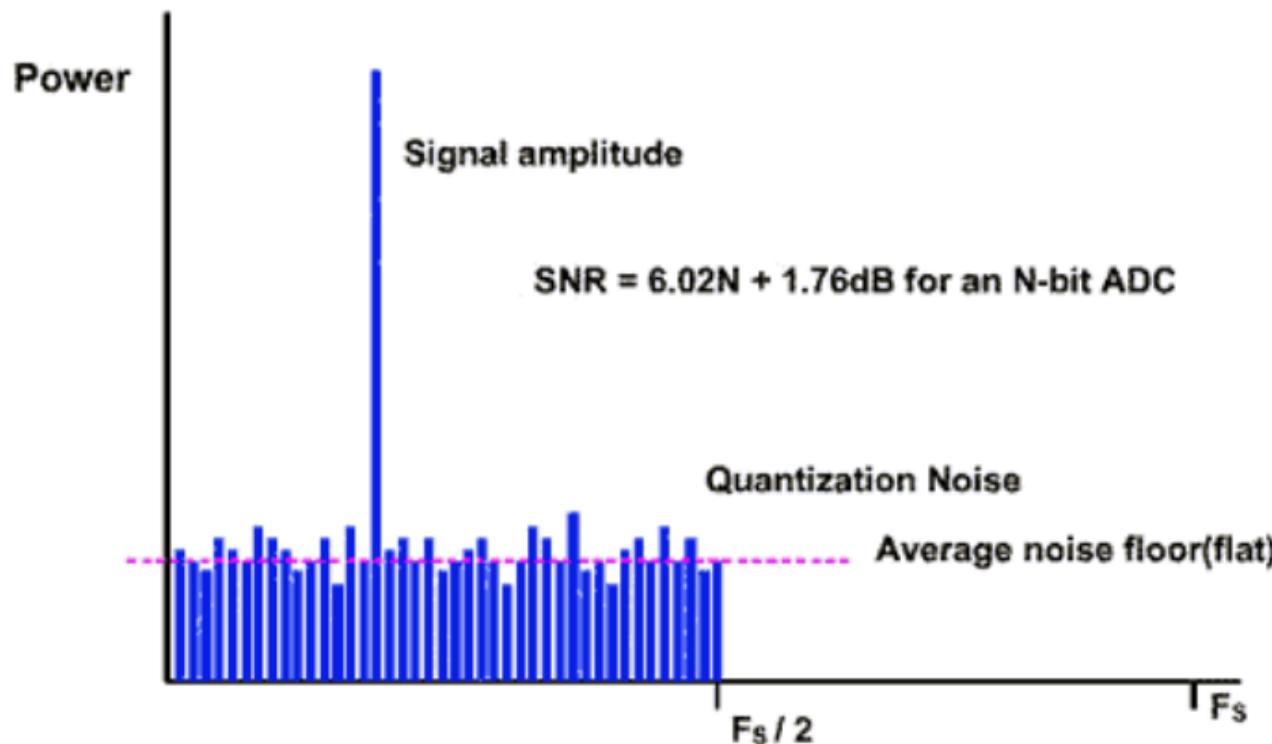
معایب : ?

تئوری Delta-Sigma ADC

به لحاظ تئوری عملکرد این ADC ها کمی پیچیده می باشد و به طور کلی مبتنی بر 4 فرآیند زیر می باشد.

- Oversampling
- Noise Shaping
- Digital filtering
- Decimation

The Frequency Domain

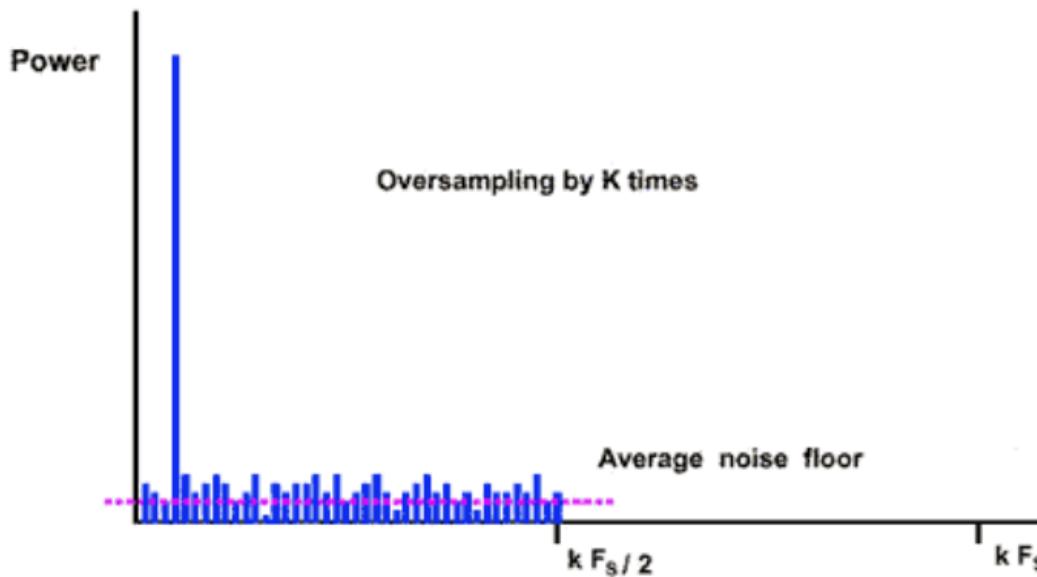


تئوری ADC

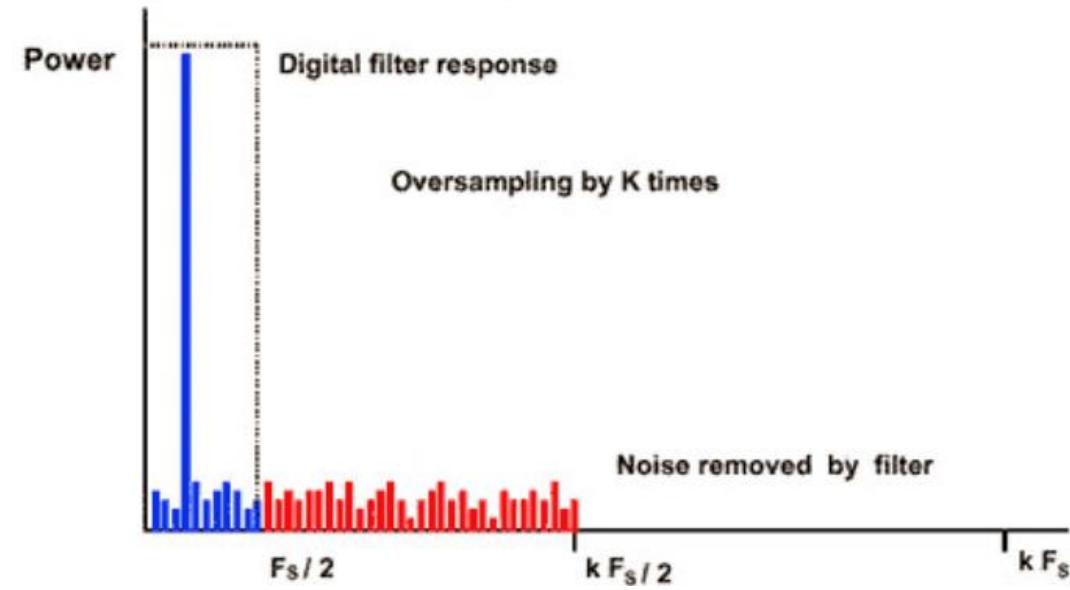
Oversampling

- با هر $4x$ افزایش در نمونه برداری، 6dB یا معادل با 1 بیت، به رزولوشن اضافه می‌شود.
- بنابراین برای دستیابی به یک ADC با رزولوشن 16 بیت، نیاز به 4^{15} بار Oversampling داریم!!! (نمی‌شود).

Oversampling by K Times



The Digital Filter

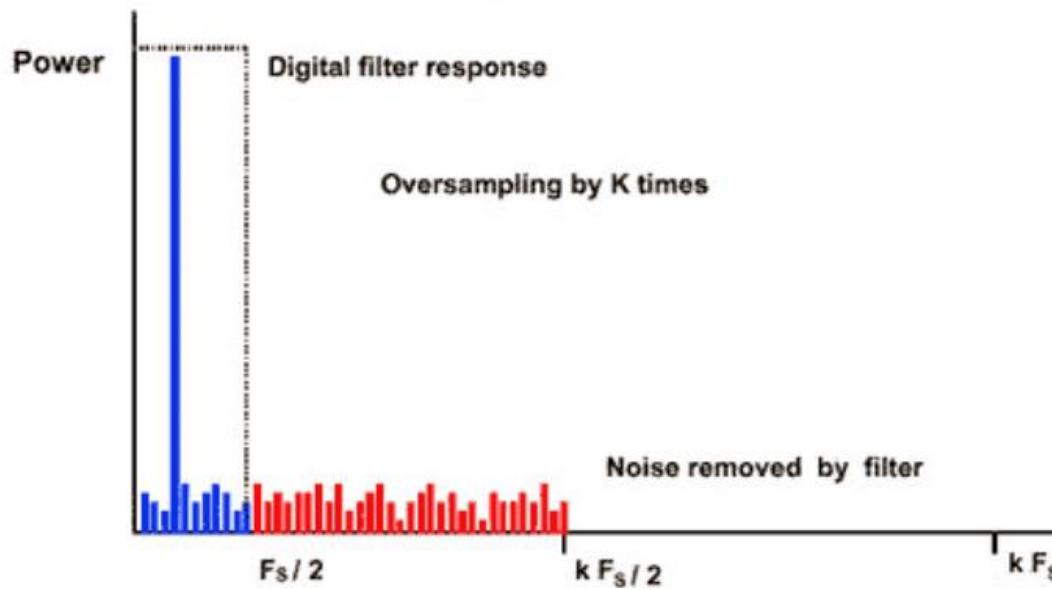


تئوری Delta-Sigma ADC

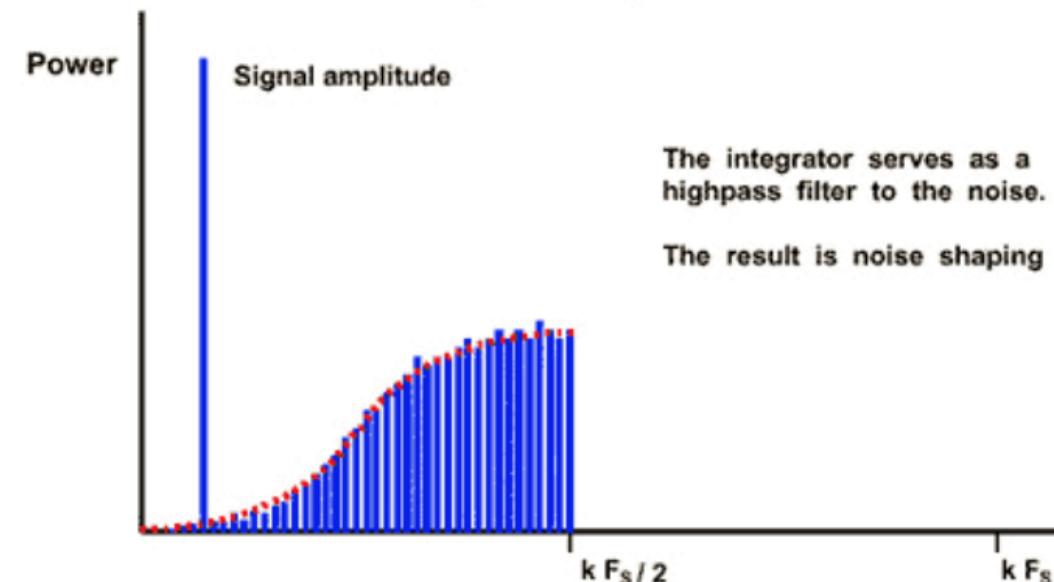
Noise Shaping ▪

▪ بار هر $2x$ افزایش در نمونه برداری، SNR 9dB به افزوده می شود. (معادل 1.5 بیت)

The Digital Filter



Noise Shaped Spectrum

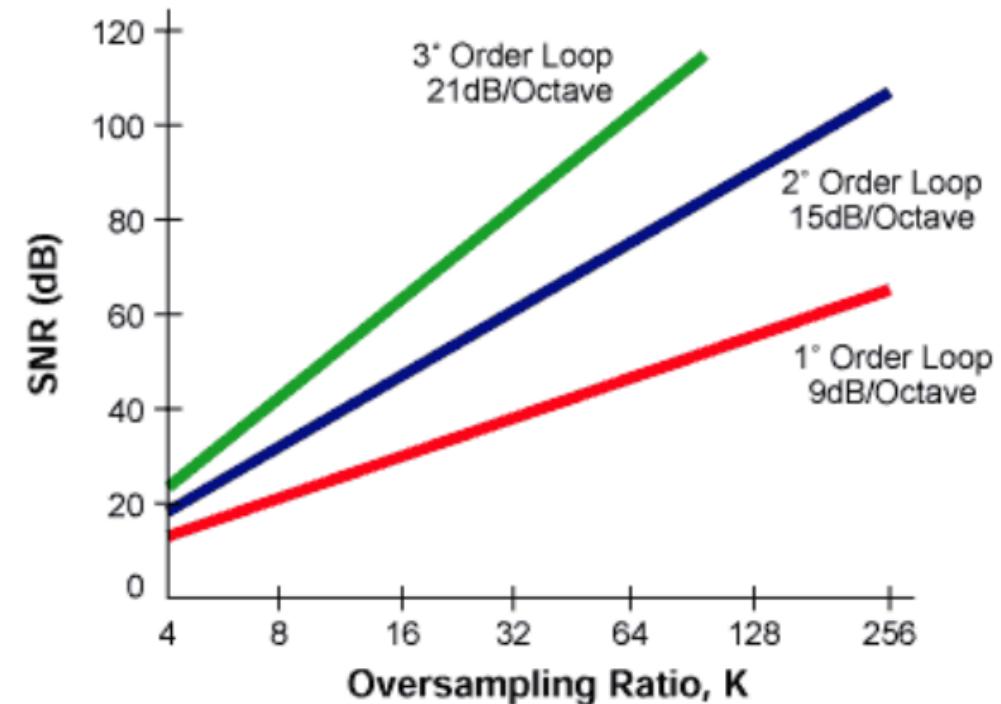
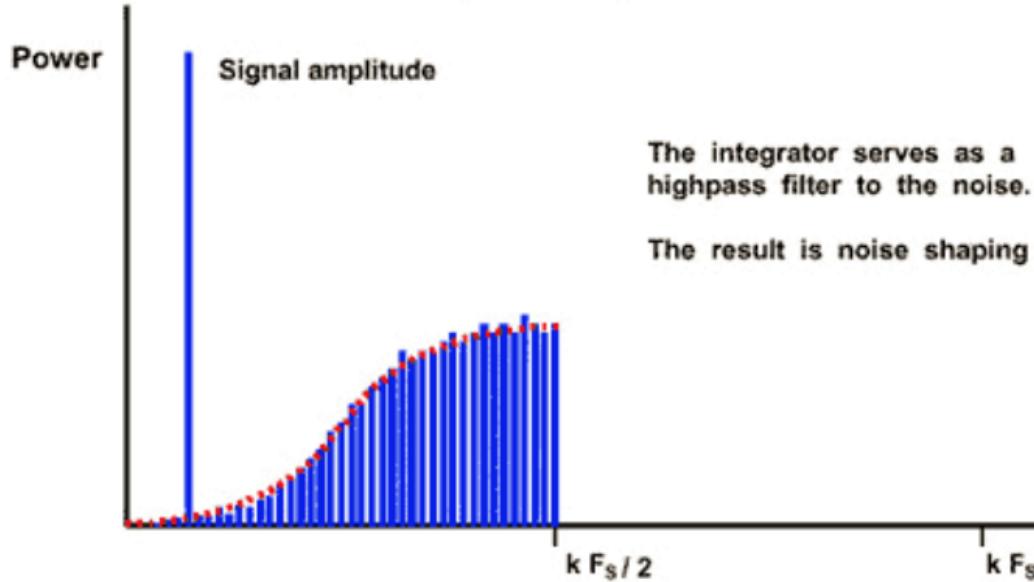


تئوری Delta-Sigma ADC

Noise Shaping

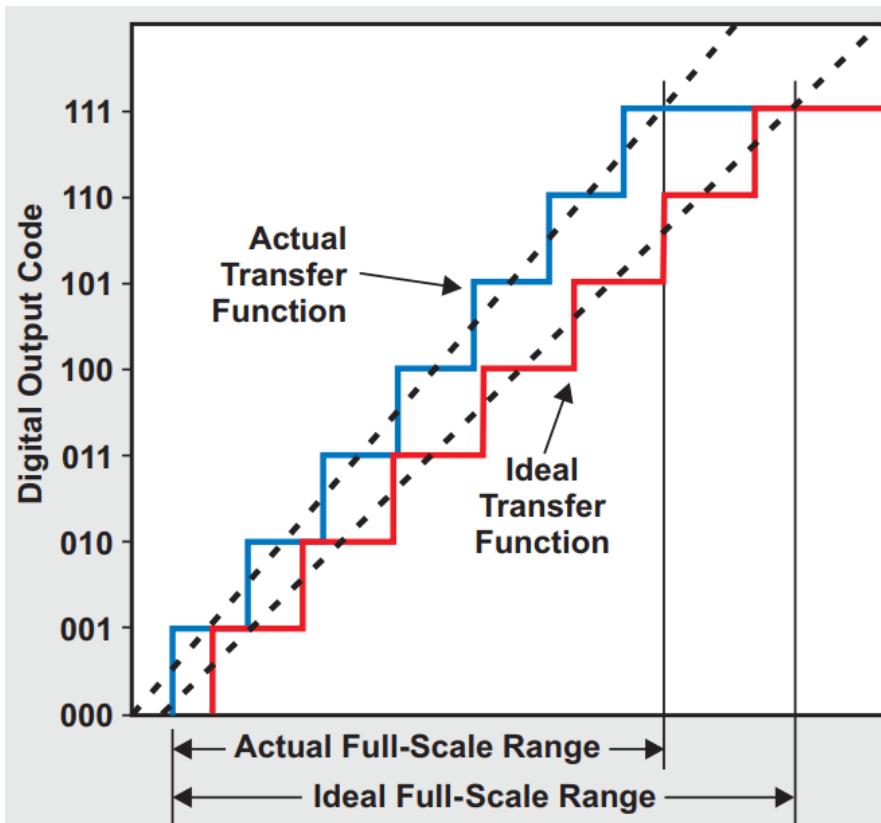
- تابع Noise Shaping نیز درجات متفاوتی دارد. در درجه 2x افزایش در سرعت نمونه برداری، افزوده می شود (معادل با 2.5 bit) و در درجه 3 حدودا 21 dB (معادل با 3.5 bit)

Noise Shaped Spectrum



ADC Static Errors

$$\text{Code} = V_{IN} \times \frac{2^n}{V_{REF}}$$



$$\text{Code} = (V_{IN} - V_{OS_ADC}) \times \frac{2^n}{V_{REF}}$$

خطای آفست!

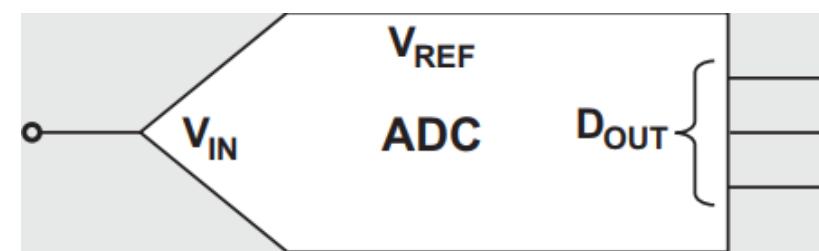
خطای Gain Error

راه حل!

$$\text{Code} = V_{IN} \times \frac{2^n}{V_{REF} (1 - GE_{ADC})}$$

$$GE_{ADC} = \frac{\text{Actual Gain} - \text{Ideal Gain}}{\text{Actual Gain}}$$

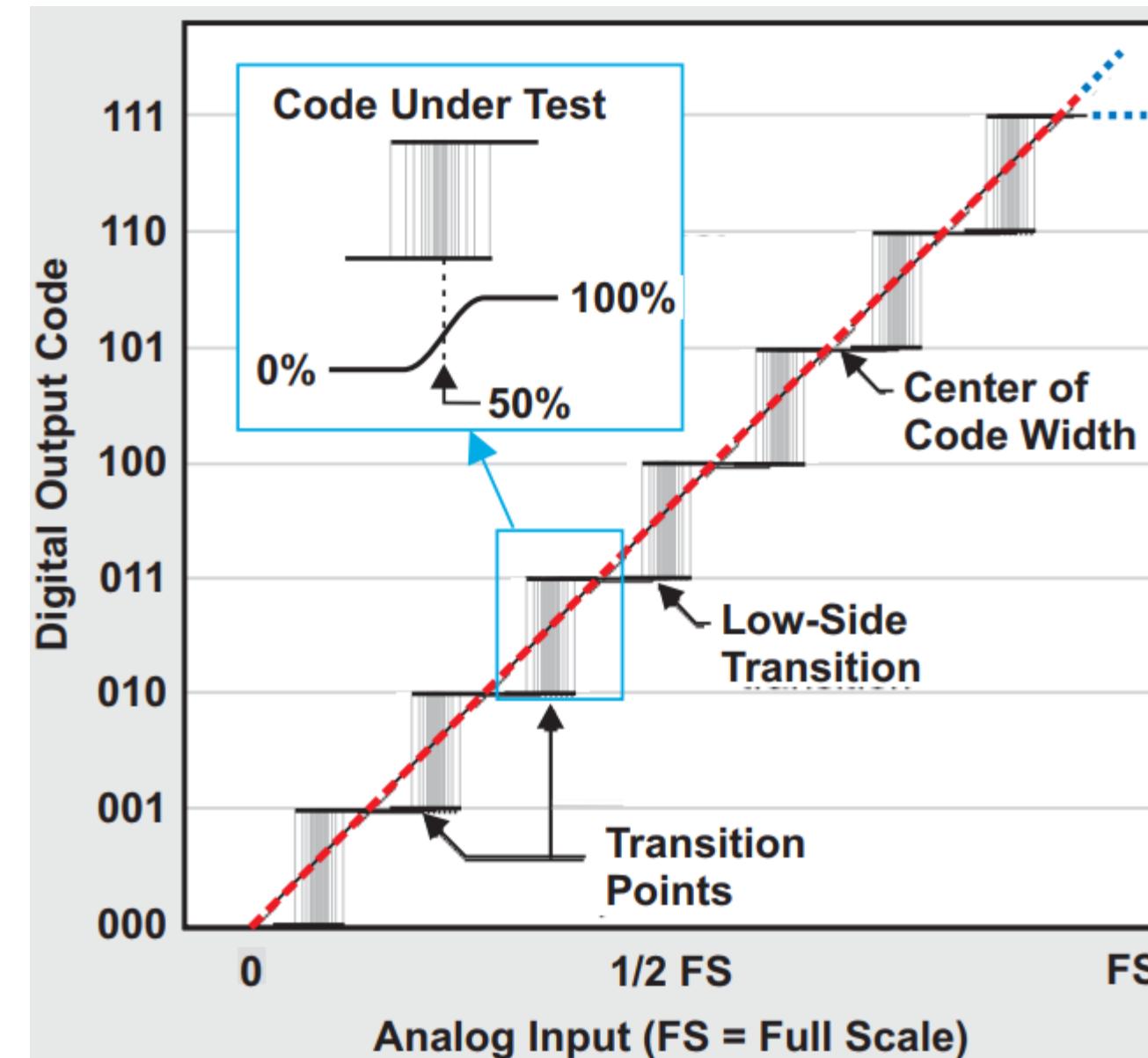
$$\text{Code} = (V_{IN} - V_{OS_ADC}) \times \frac{2^n}{V_{REF} (1 - GE_{ADC})}$$



ADC Static Errors

- خطای انتقال از یک سطح به سطح دیگر معمولاً به دلیل عدم تصمیم‌گیری صحیح Comparator ها می‌باشد.

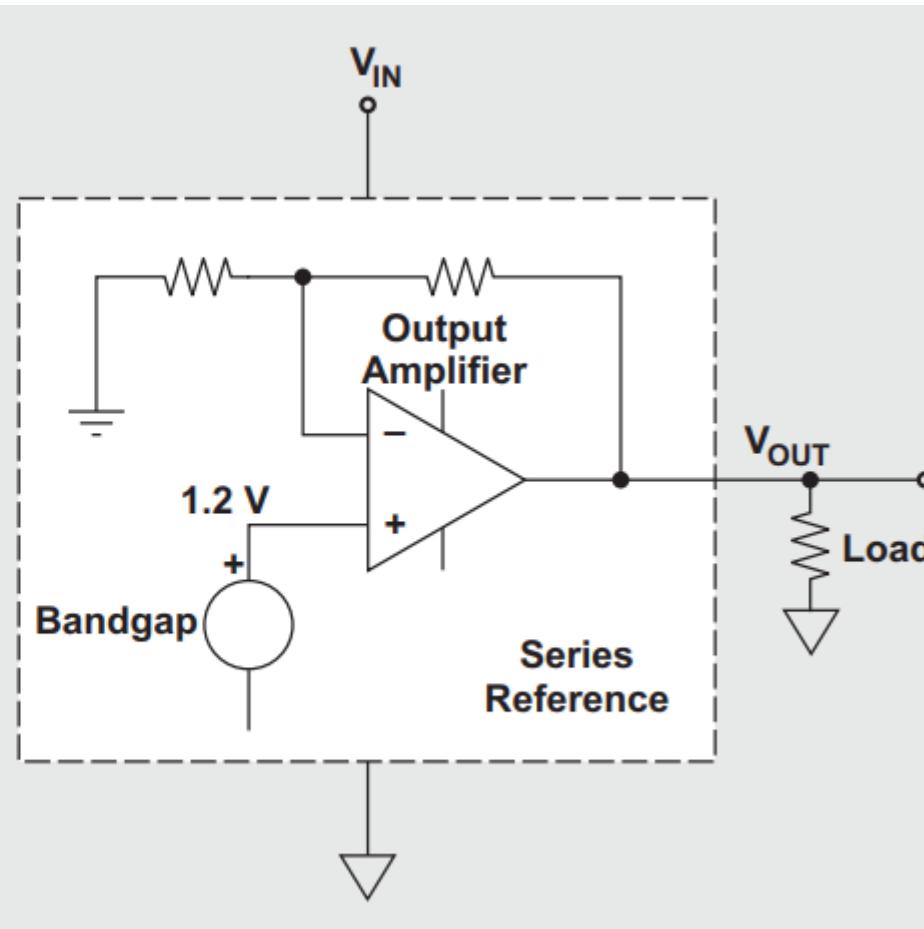
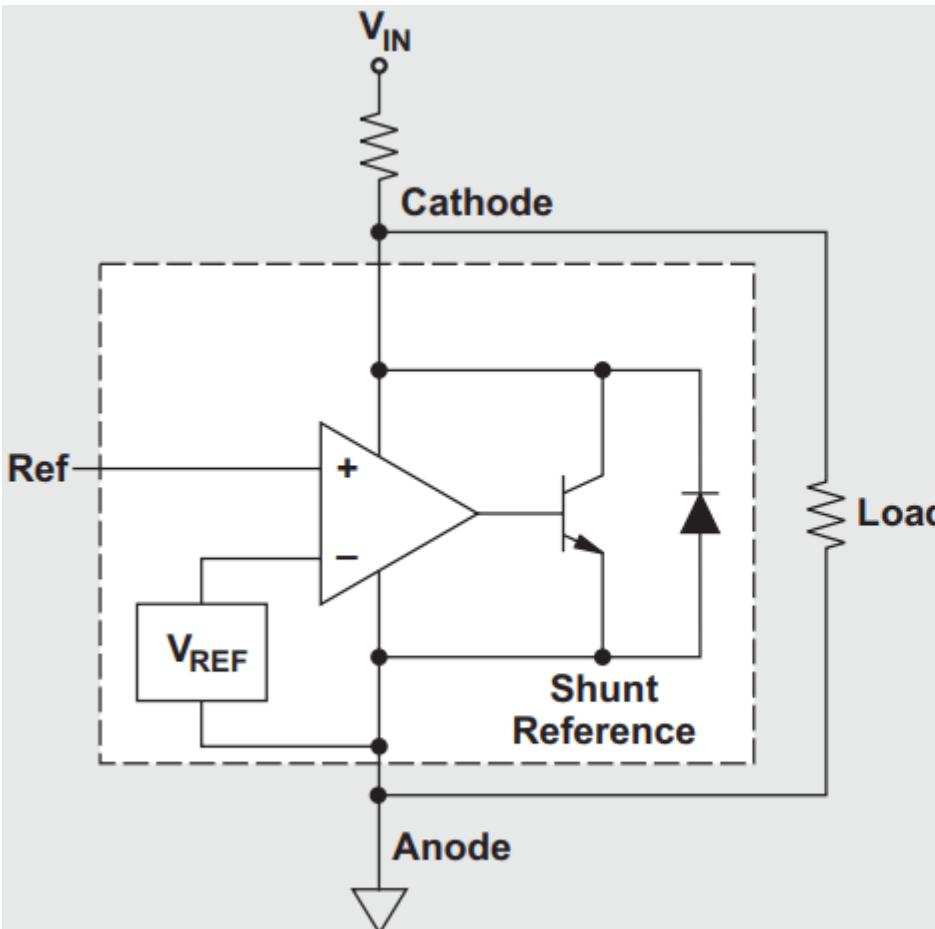
- عوامل موثر بر خطا در تصمیم‌گیری Comparator ها می‌تواند ناشی از خطاهای داخلی نظیر مشخصه غیرخطی، آفست و ... باشد. همچنین عوامل خارجی نظیر نویز محیطی، سیگنال‌های القایی، تغییرات دما منجر به تغییرات پارامترهای Comparator می‌باشند.



انتخاب ولتاژ رفرنس مناسب

- تغییرات ولتاژ شروع بین 0.5% تا 5%
- تغییرات ولتاژ بر حسب دما به طور معمول بین 50 تا $100\text{uV}^{\circ}\text{C}$
- این ساختار قابل پیاده‌سازی ولتاژهای مثبت، منفی می‌باشد. (همچنین پایه Ref می‌تواند Float باشد.)

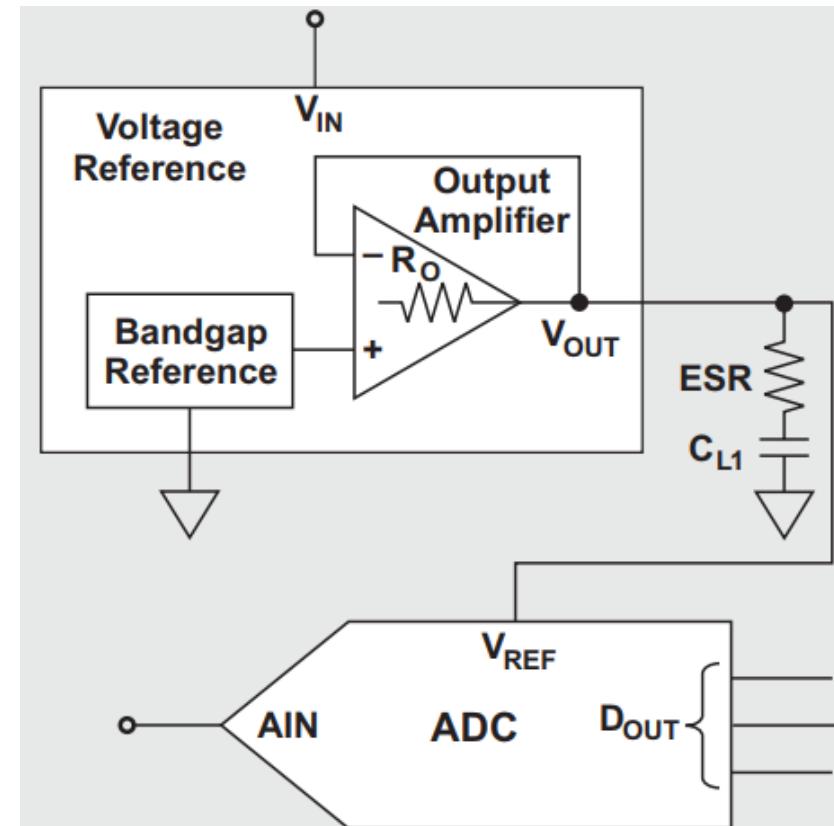
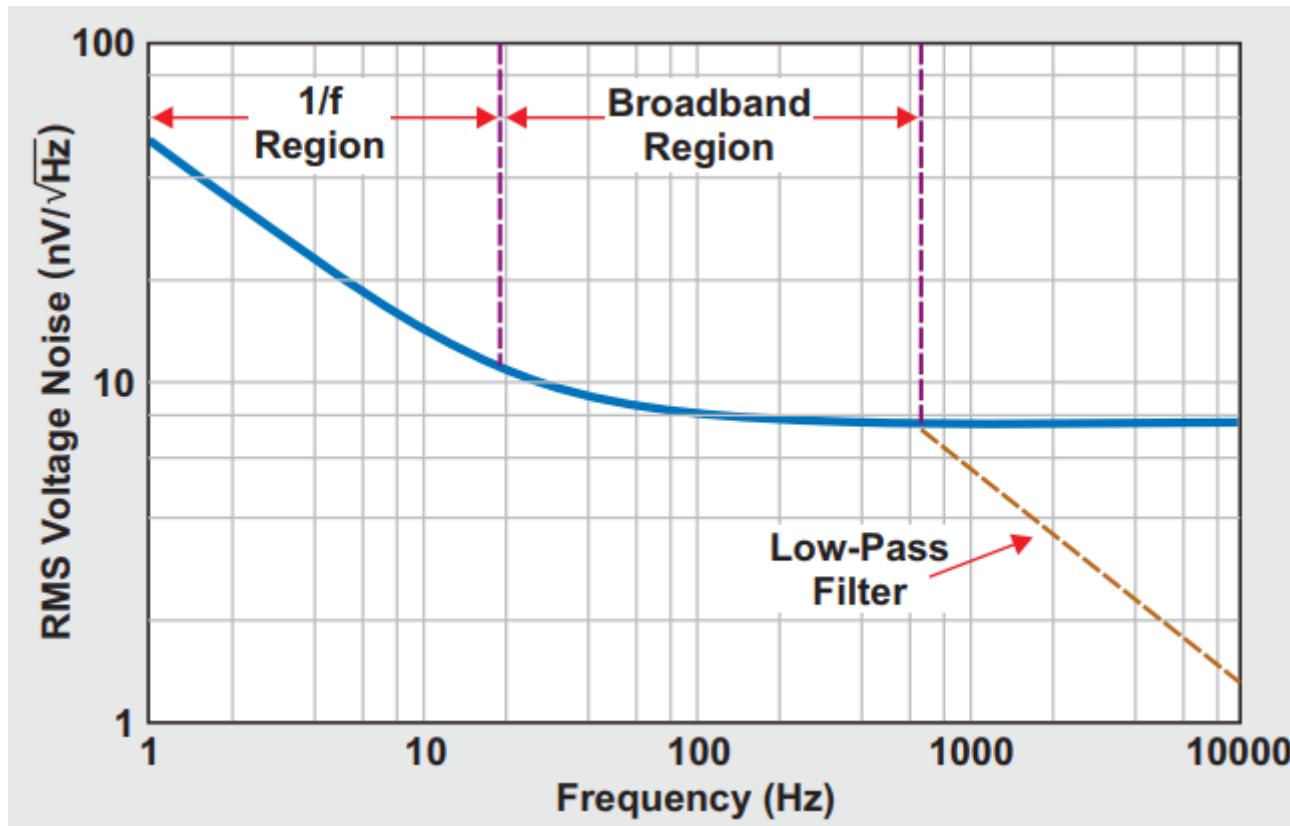
- تفاوت ولتاژ رفرنس و رگولاتور؟



- تغییرات ولتاژ شروع بین 0.5% تا 0.05%
 - تغییرات ولتاژ بر حسب دما به طور معمول بین $2.5 \text{ ppm}^{\circ}\text{C}$
 - این ساختار معمولاً ولتاژ خارجی ثابتی می‌دهد
- سوال : از چه تعداد رزولوشن به بعد بایستی از رفرنسهای سری استفاده کرد؟

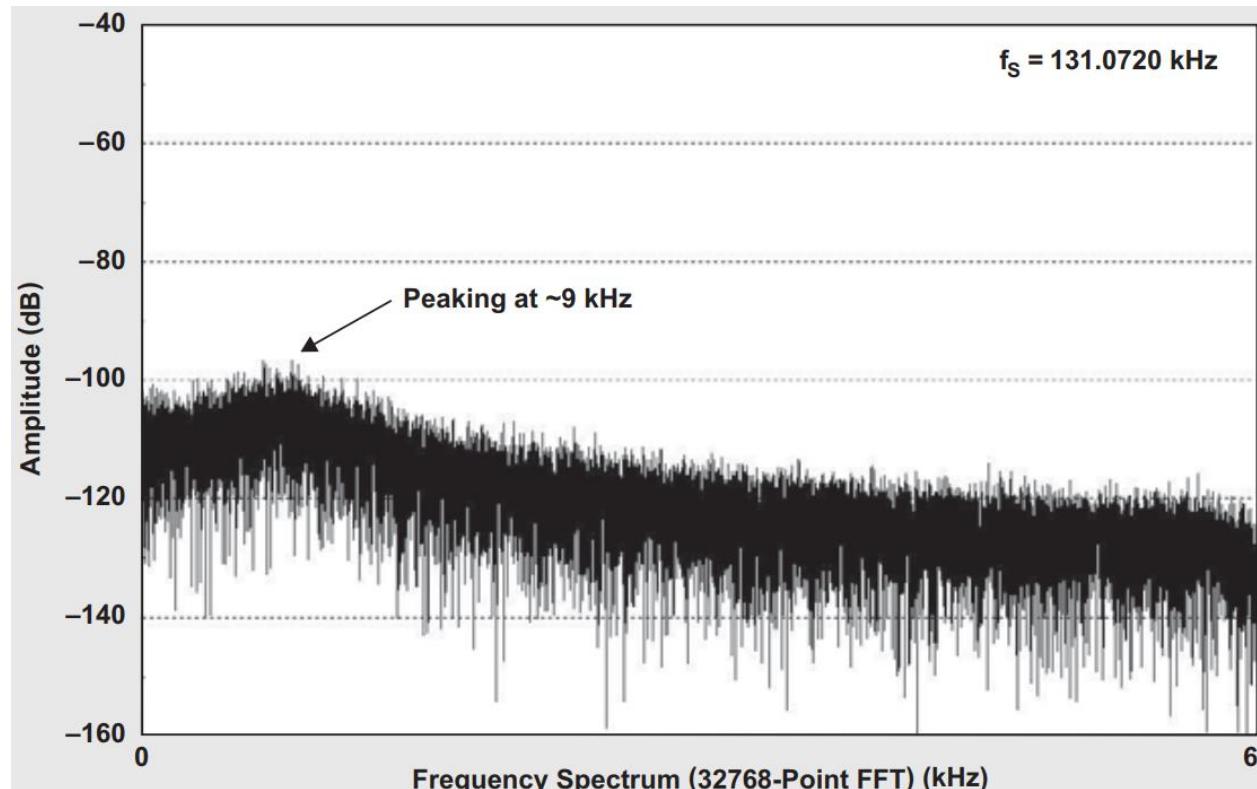
نویز ولتاژ رفرنس

- افزودن یک خازن در خروجی، می‌تواند اثر نویز ولتاژ رفرنس را فیلتر نماید.
- چه خازنی؟! چه جنسی؟! چه اندازه‌ای؟! (مفهوم ESR توضیح داده شود)



نویز ولتاژ رفرنس برای ADC‌های با رزولوشن بالا

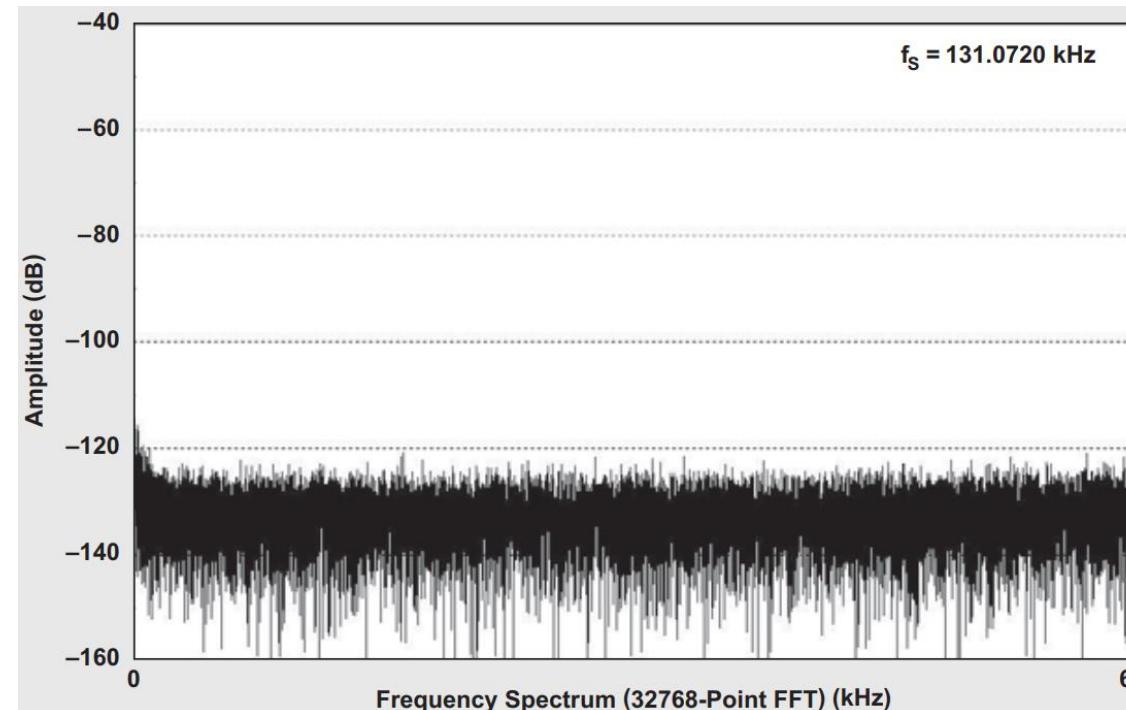
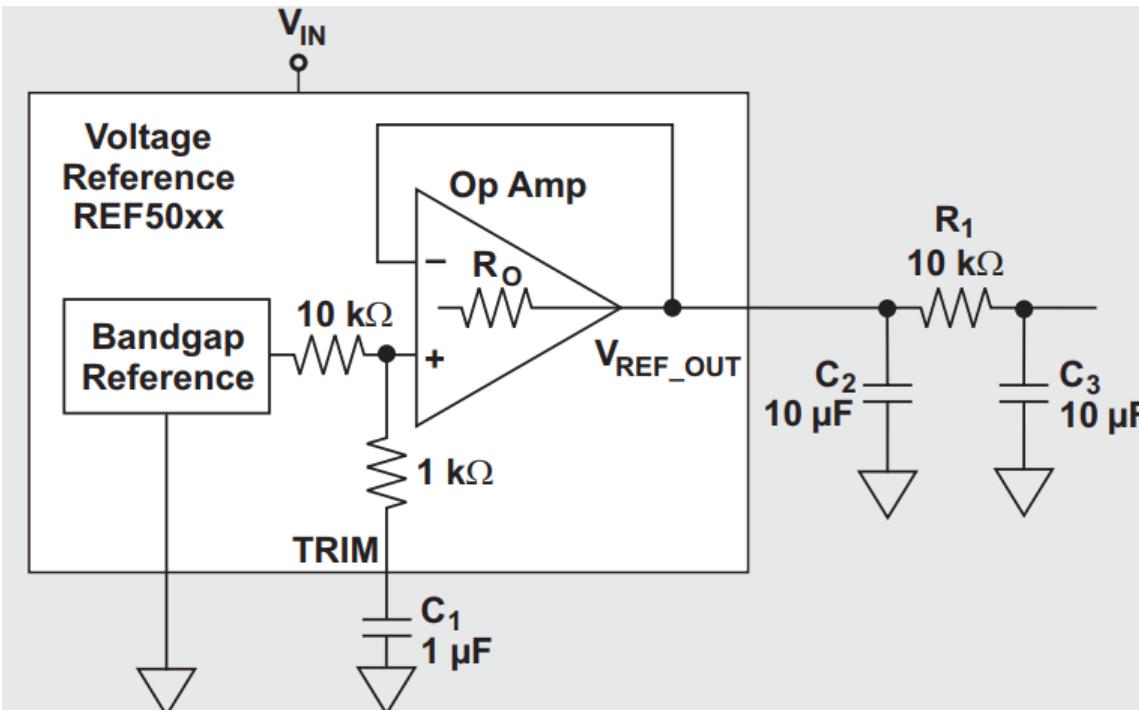
- مثال : شکل مقابل نویز منبع تغذیه را نشان می‌دهد.
اندازه‌گیری‌ها نشان می‌دهد که این نویز (قریبا 138 μV p-p) است و برای ADC‌های با رزولوشن کمتر از 14 بیت مناسب می‌باشد!؟



اما برای ADC‌های با رزولوشن بالاتر، می‌بایست نویز فرکانس
های پایین را نیز، کاهش دهیم (استفاده از فیلتر RC).

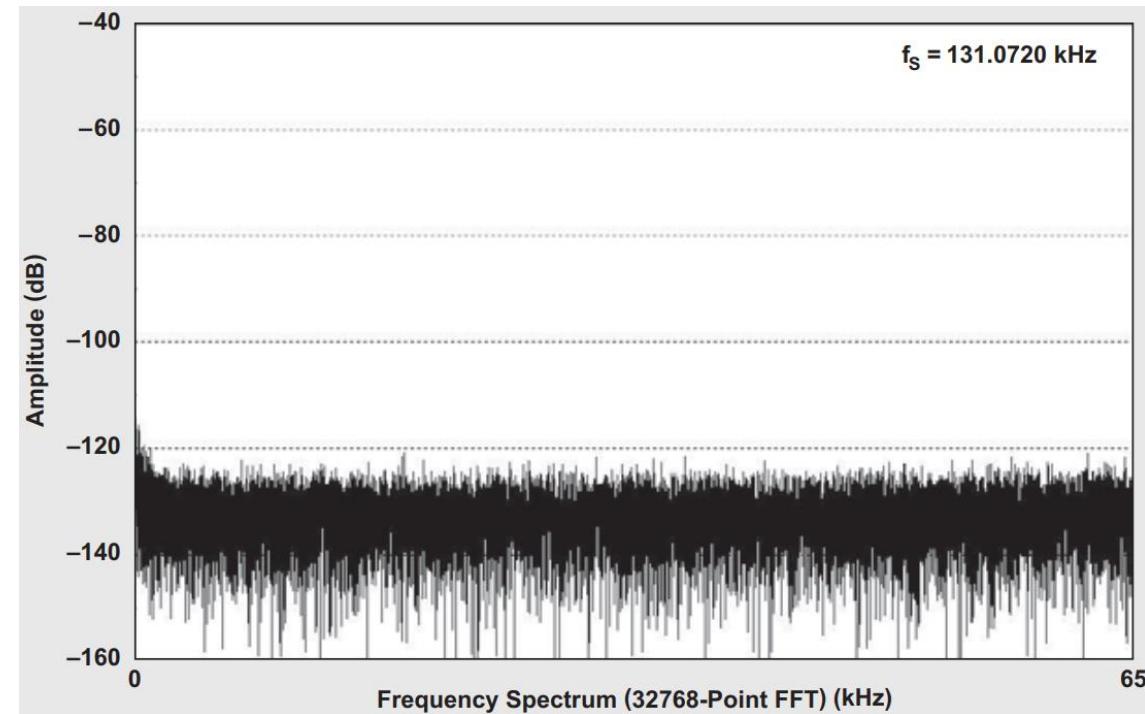
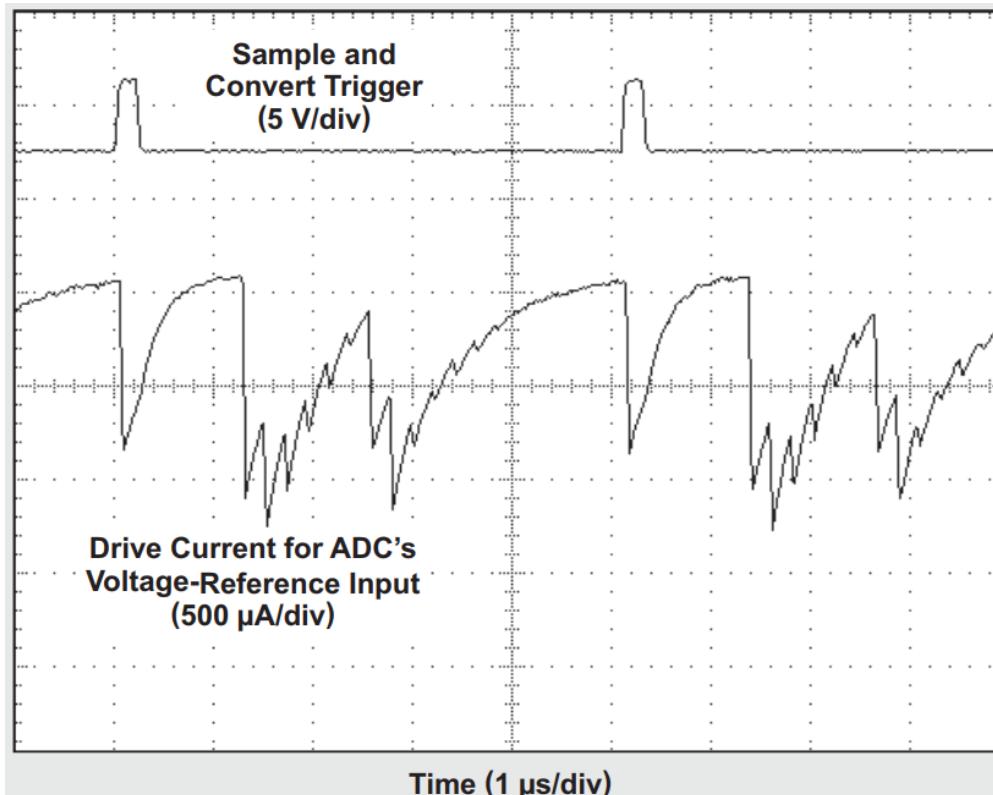
کاهش نویز تغذیه برای ADC های +16 بیت

- استفاده از یک فیلتر پایین‌گذر با فرکانس قطع نزدیک به فرکانس ۰ (مثلا ۱.۵۹ Hz)
- با توجه به دیاگرام نویز، سطح نویز در فرکانس‌های پایین نیز بطور چشمگیری کاهش می‌یابد. اندازه‌گیری‌ها نشان می‌دهد که مقدار تقریبی نویز پس از افزودن فیلتر RC به مدار قبل، به ۱۵uv P-P کاهش یافته است (حدودا ۹۰% بهبود).

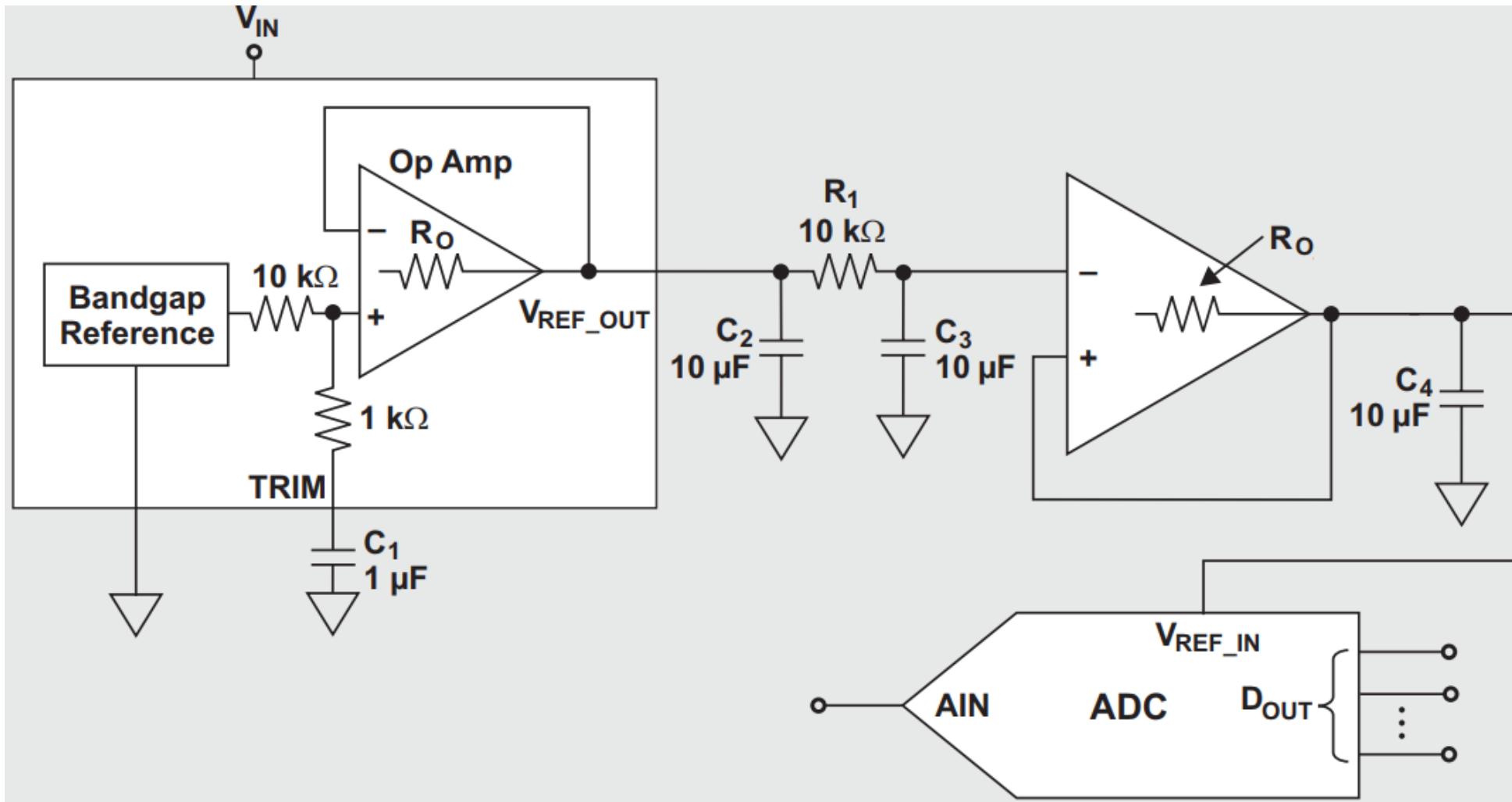


کاهش نویز تغذیه برای ADC های +16 بیت

- با افزودن مقاومت در مسیر ولتاژ رفرنس، هنگامی که ADC شروع به نمونه برداری می کند، جریانی از مقاومت می گذرد که سبب اختلاف پتانسیل و اووجاچ تغذیه می شود. هر چند سطح نویز ولتاژ رفرنس بهبود یافته است، اما تغییرات ولتاژ به سبب جریان کشی ADC افزوده شده است. به منظور رفع این مشکل، بایستی یک Buffer به خروجی فیلتر RC اضافه نماییم تا جریان ADC توسط خروجی تامین گردد.



ساختار نهایی تغذیه برای ADC های +16 بیت



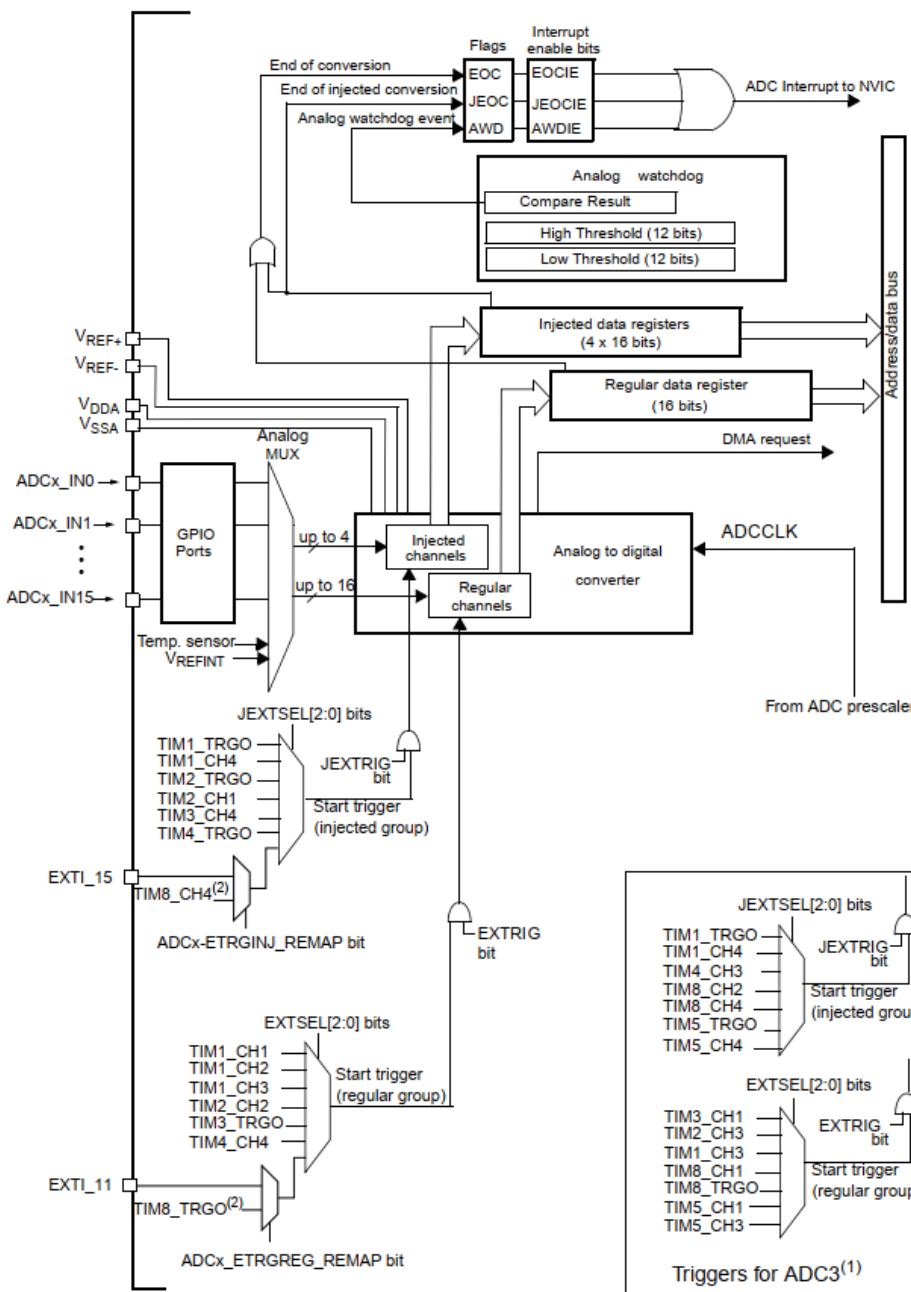
STM32F10X ADC

- 12 بیت رزولوشن (در خانواده‌های F0 قابلیت تنظیم تعداد بیت نیز وجود دارد.
- Multiplex 16 کanal به صورت
- قابلیت تولید وقفه در
- انتهای نمونه برداری کanal regular
- انتهای نمونه برداری کanal injected
- Analog watchdog عبور از مرزهای تعیین شده
- قابلیت نمونه برداری تکی یا متوالی
- قابلیت تعریف سناریو نمونه برداری از کanal 0 تا n
- کالibrاسیون داخلی
- قابلیت تنظیم زمان نمونه برداری مجزا برای هر کanal

STM32F10X ADC

- قابلیت راهاندازی به نمونه برداری داخلی و خارجی (توسط سایر واحدها)
- قابلیت نمونه برداری Discontinues
- قابلیت پردازنده‌های با بیشتر از یک ADC (Dual Mode)
- قابلیت تنظیم Clock و سرعت نمونه برداری (حداکثر 1 MSPS)
- تغذیه مورد نیاز ADC بین 2.4 تا 3.6 ولت
- ولتاژ ورودی بین $+V_{ref}$ - $-V_{ref}$

بلوک دیاگرام ADC



- در ساختار STM32f103، پایه Vref+ در دسترس کاربر نمی‌باشد و به صورت داخلی به پایه VDDA متصل شده است. همچنین پایه Vref- نیز به پایه VSSA متصل می‌باشد.
- در پردازنده‌هایی که Vref- در اختیار کاربر می‌باشد، بایستی حتماً بصورت خارجی به VSSA متصل گردد.

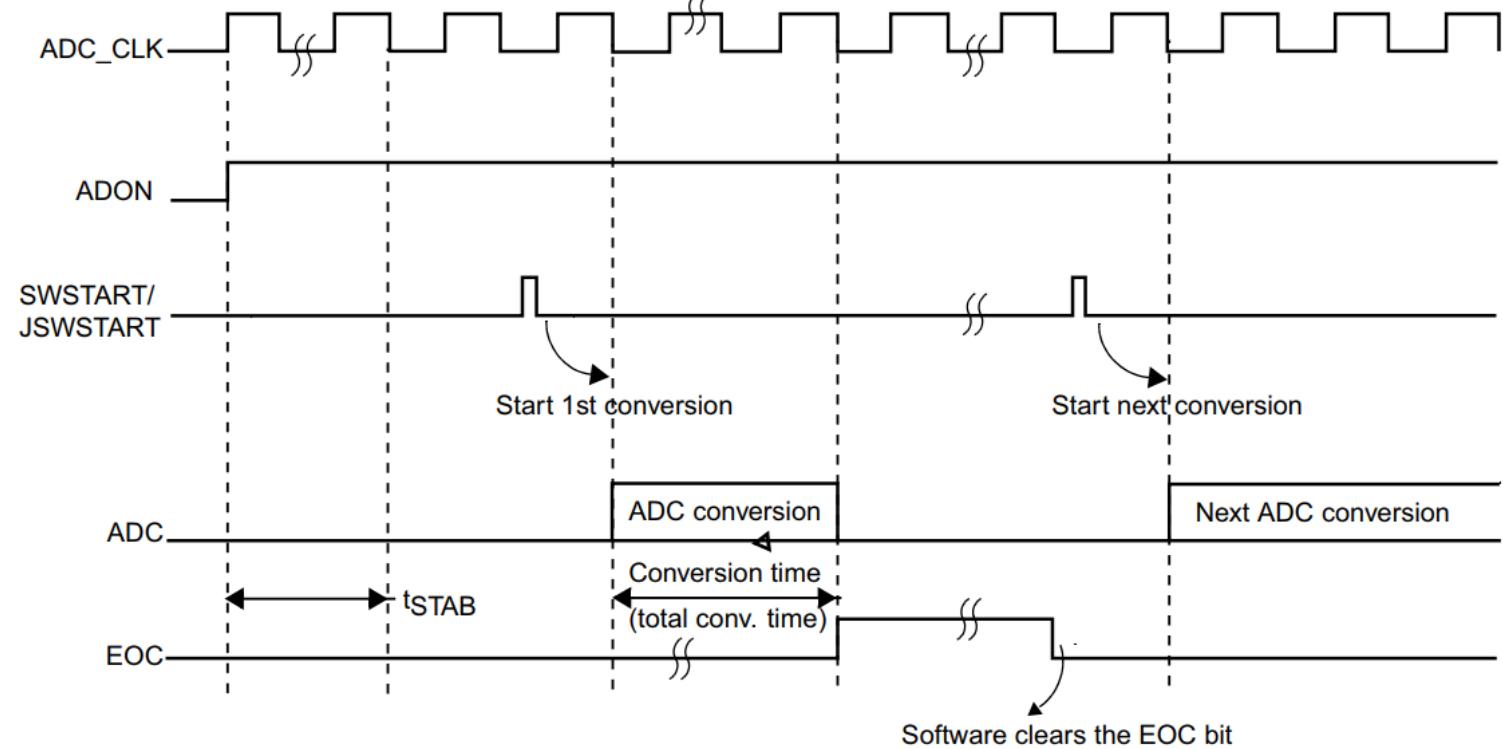
Single Conversion VS Continues Conversion

در Single Conversion، پس از اتمام نمونه‌گیری، بیت EOC یک می‌شود تا زمانی که کاربر آن را بخواند. سپس می‌توان نمونه‌گیری بعدی را آغاز کرد.

در Continues Conversion، بلافاصله پس از اتمام هر نمونه‌گیری، نمونه‌گیری بعد آغاز می‌شود (مشابه با حالت قبل، در پایان هر نمونه، بیت EOC یک می‌شود).

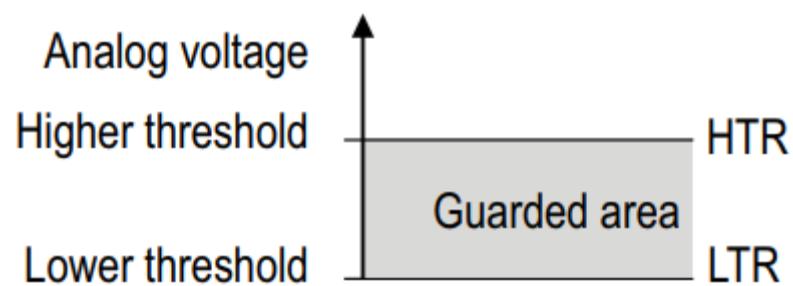
قابلیت تعريف وقفه برروی بیت EOC وجود دارد.

توجه شود که در هنگام روشن نمودن واحد ADC، برای بار نخست، بایستی اندکی صبر نمایید (tstab).



Analog Watchdog

- یکی از ویژگی‌های ADC در میکروکنترولرهای STM32، قابلیت Analog Watchdog می‌باشد. در این قابلیت می‌توان دو آستانه بالا (Higher threshold) و آستانه پایین (Lower threshold) تعریف نمود. سیگنال ورودی همواره توسط این بخش در حال ارزیابی است و مادامی که دامنه سیگنال در محدوده آستانه پایین تا آستانه بالا باشد، پیغامی صادر نمی‌شود. اما به محض آن که مقدار ورودی از آستانه بالا فراتر و یا از آستانه پایینی، پایین‌تر رود، بیت مخصوص به هر کدام یک می‌شود. همچنین وقفه متناظر با هر یک نیز اجرا می‌گردد.



Scan mode

- این قابلیت وجود دارد که به کانال‌های مختلف یک ADC، چندین سنسور مستقل متصل نمود و آن‌ها در یک آرایه ترتیبی مرتب نمود. در این حالت با هر بار ارسال دستور نمونه‌برداری، مقدار هر کانال به ترتیب توالی از پیش تعیین شده توسط کاربر خوانده و در رجیستر ADC_DR قرار داده می‌شود. این عملیات تا آخرین کانال ادامه می‌یابد.
- مثال : فرض کنید قصد دارید در یک گلخانه پارامترهای دما، رطوبت، گاز CO₂ و اکسیژن را اندازه‌گیری نمایید. در این راستا سنسورهای مذکور را به 4 کانال دلخواه از ADC متصل نموده و ترتیب آن‌ها را برنامه‌ریزی می‌کنیم. سپس ADC را روشن و نمونه‌گیری را آغاز می‌کنیم. (فرض کنید ترتیب فوق معتبر باشد). ADC پس از شروع به کار نخست از سنسور دما نمونه‌برداری می‌کند. بلافاصله پس از آن مقدار سنسور رطوبت پس از آن CO₂ و در انتهای نیز اکسیژن را اندازه‌گیری می‌کند.
- توجه شود که برای استفاده از قابلیت Scan mode باستی از بلوک DMA کمک بگیریم! (راهی برای دور زدن وجود دارد!?)
- را می‌توان در هر دو ساختار single یا Continues راهاندازی کرد.

Discontinues mode

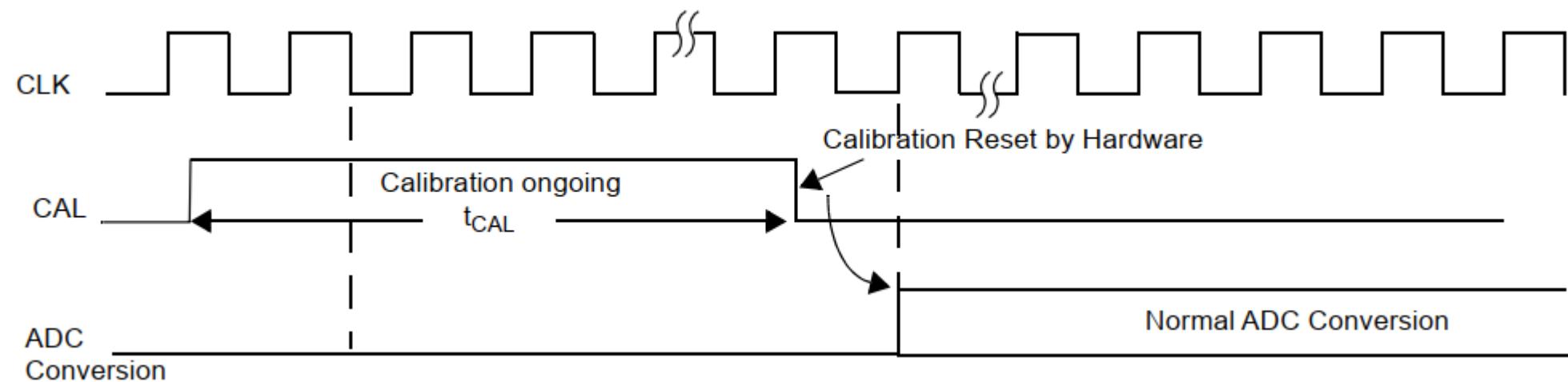
- مشابه با حالت Scan mode است با این تفاوت که در این قابلیت می‌توان تعداد کانال‌هایی که در هر بار تحریک ADC نمونه‌برداری می‌شوند را تعیین نمود (channel number ≤ 8).
- مثال : فرض کنید به کانال‌های 0, 1, 2, 3, 6, 7, 9, 10 سنسورهای مختلف متصل نمودیم و قصد نمونه‌برداری داریم. با فرض (channel number = 3) :
- تحریک اول : کانال‌های 0, 1, 2 نمونه‌برداری می‌شود.
- تحریک دوم : کانال‌های 3, 6, 7 نمونه‌برداری می‌شود.
- تحریک سوم : کانال‌های 9, 10 نمونه‌برداری می‌شود.
- تحریک چهارم : ?!

Regular channels VS Injected Channels

- برای تمامی سناریوهای Injected و ... می‌توان کانال‌ها را از نوع Discontinues mode ،Scan mode یا regular و چند سنسور از نوع Injected تعريف نمود.
- کانال Injected همان کانال معمولی است با اولویت بالاتر، به این معنی که اگر چند سنسور از نوع regular و چند سنسور از نوع Injected تعريف شده باشند. اولویت نمونهبرداری با کانال‌های Injected می‌باشد. فرض کنید مقادیر سنسورهای متصل به کانال‌های Regular در حال خواندن می‌باشد که سیگنال تحریک نمونهبرداری کانال‌های Injected صادر می‌شود. در این شرایط عملیات نمونهبرداری از سنسورهای کانال regular متوقف شده و نمونهبرداری سنسورهای متصل به کانال Injected انجام می‌شود و پس از پایان آن، ادامه عملیات نمونهبرداری از کانال‌های regular آغاز می‌شود.

Calibration

- در پردازنده‌های STM32 قابلیت کالیبراسون داخلی وجود دارد (چگونه؟!). پس از روشن کردن بلوک ADC، می‌توان با ارسالی دستوری، این بلوک را در وضعیت کالیبراسیون قرار دهیم. برای انجام نمونه‌گیری بایستی منتظر بمانیم تا زمان کالیبراسیون به اتمام برسد (t_{cal}).



قابلیت تنظیم زمان نمونهبرداری مجزا برای هر کanal

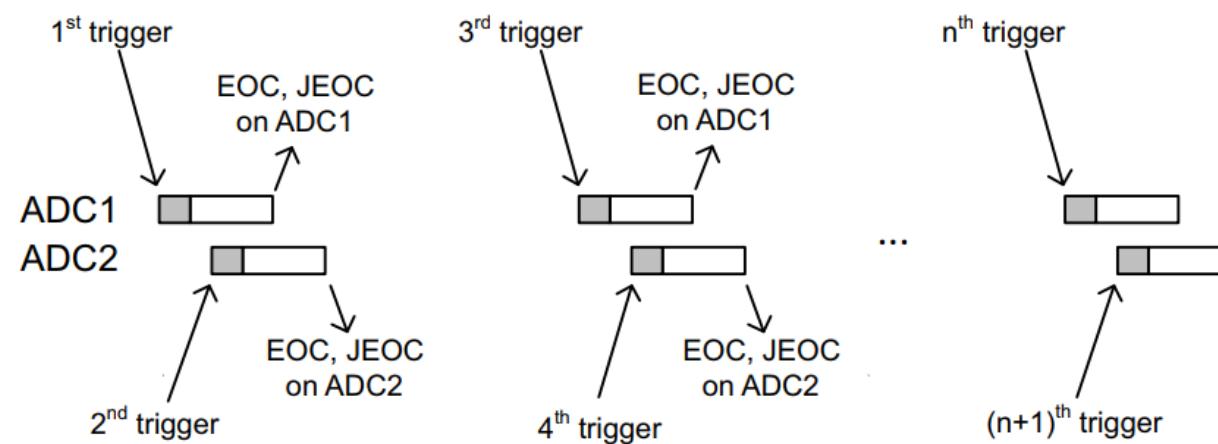
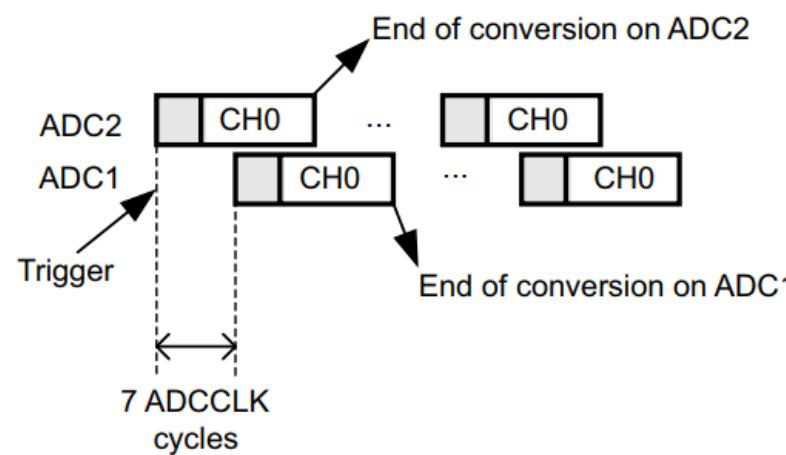
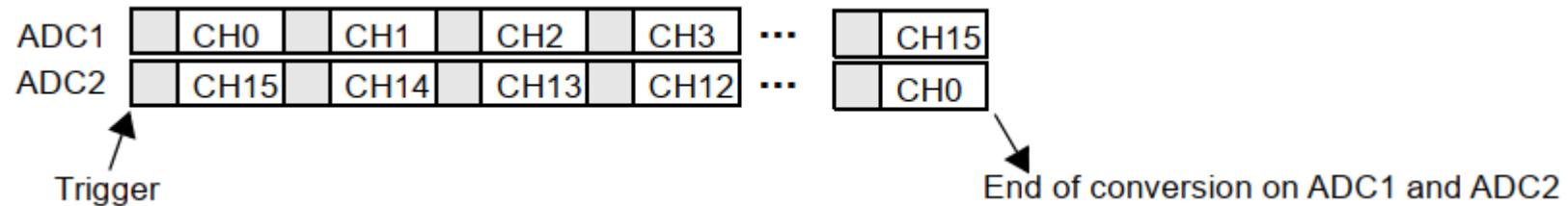
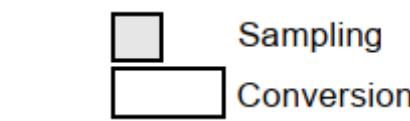
- قابلیت تنظیم زمان نمونهبرداری برای هر کanal به صورت مجزا (چرا!!)

$T_{CONV} = Sampling\ time + 12.5\ cycles$ زمان نمونهبرداری از رابطه زیر پیروی می‌کند :

- مثال : با فرض فرکانس ADC Clock = 14 MHz، زمان نمونه برداری ADC = 1.5 us

$$T_{CONV} = 1.5 + 12.5 = 14\ cycles = 1\ us$$

Dual ADC mode



- Injected simultaneous mode
- Regular simultaneous mode
- Fast Interleaved mode
- Slow interleaved mode
- Alternate trigger mode
- Independent mode

توابع HAL مورد نیاز برای راهاندازی ADC

`HAL_StatusTypeDef HAL_ADC_Start (ADC_HandleTypeDef * hadc)`

`HAL_StatusTypeDef HAL_ADC_Stop (ADC_HandleTypeDef * hadc)`

`HAL_StatusTypeDef HAL_ADC_PollForConversion (ADC_HandleTypeDef * hadc, uint32_t Timeout)`

`HAL_StatusTypeDef HAL_ADC_Start_IT (ADC_HandleTypeDef * hadc)`

`HAL_StatusTypeDef HAL_ADC_Stop_IT (ADC_HandleTypeDef * hadc)`

`HAL_StatusTypeDef HAL_ADC_Start_DMA (ADC_HandleTypeDef * hadc, uint32_t * pData, uint32_t Length)`

`HAL_StatusTypeDef HAL_ADC_Stop_DMA (ADC_HandleTypeDef * hadc)`

`uint32_t HAL_ADC_GetValue (ADC_HandleTypeDef * hadc)`