

11_conversion_ad_da_cad

June 19, 2026

1 Conversión Analógica-Digital / Digital-Analógica y diseño asistido (CAD)

Electrónica Digital — 2.º GIERM (Mecatrónica)

Apuntes propios destilados de los temas oficiales: - *Tema 11 — Conversión Digital-Analógica (DAC a frecuencia de Nyquist)* - *Tema 12 — Conversión Analógico-Digital (ADC a frecuencia de Nyquist)* - *Tema 12.2 — Sistemas de conversión sobremuestreados (Sigma-Delta)* - *Tema 12 — Nociones de CAD / flujo de diseño digital*

Cómo usar este notebook (enfoque examen): cada bloque empieza con el resultado/fórmula clave (worked example), luego la justificación, y termina con una **figura** generada con **numpy/matplotlib** que ilustra el concepto. Es autocontenido: ejecútalo de arriba a abajo, solo necesitas **numpy** y **matplotlib**.

1.1 Índice

1. Cadena de adquisición de señal
2. Muestreo y teorema de Nyquist
3. Aliasing (solapamiento espectral)
4. Cuantización y error de cuantización
5. Relación señal-ruido (SNR) y ENOB
6. DAC: idea, peso binario y red R-2R
7. Curvas de transferencia y errores estáticos (offset, ganancia, DNL, INL)
8. ADC Flash (paralelo)
9. ADC de aproximaciones sucesivas (SAR)
10. ADC de rampa (simple y doble)
11. Convertidores sobremuestreados: Sigma-Delta
12. CAD: flujo de diseño digital
13. Resumen y chuleta de examen

Entorno listo: 1.26.4

1. Cadena de adquisición de señal

El mundo físico es **analógico** (continuo en tiempo y amplitud); el procesado se hace en **digital** (número finito de estados, discreto en tiempo y amplitud). La cadena típica es:

```
x(t)          +-----+   +-----+   +-----+   x[n]   +-----+   y[n]   +-----+   +-----+
analógica -> | AAF | -> | S&H | -> | ADC | -----> | Proceso | -----> | DAC | -> | LPF rec
          +-----+   +-----+   +-----+   digital +-----+   digital +-----+   +-----+
```

filtro muestreo cuantiza
anti-alias y retencion y codifica

reconstrucción

- **AAF (filtro anti-aliasing):** limita el ancho de banda antes de muestrear (evita aliasing).
- **S&H (Sample & Hold):** congela la tensión durante la conversión. *Todos* los ADC lo necesitan: la entrada debe ser **constante** mientras el ADC decide.
- **ADC:** genera un código de n bits a partir de la entrada analógica.
- **DAC:** genera una tensión analógica proporcional al código digital, relativa a una referencia V_{ref} .
- **LPF de reconstrucción:** suaviza la salida escalonada del DAC.

Relación fundamental DAC (lo mismo “al revés” para el ADC):

$$V_{out} = V_{ref} \cdot \frac{D}{2^n} \quad \Leftrightarrow \quad D = \text{round}\left(2^n \cdot \frac{V_{in}}{V_{ref}}\right)$$

donde D es el código ($0 \dots 2^n - 1$) y $q = V_{ref}/2^n$ es el **escalón de cuantización** (LSB).

2. Muestreo y teorema de Nyquist

Worked example primero — el resultado a memorizar:

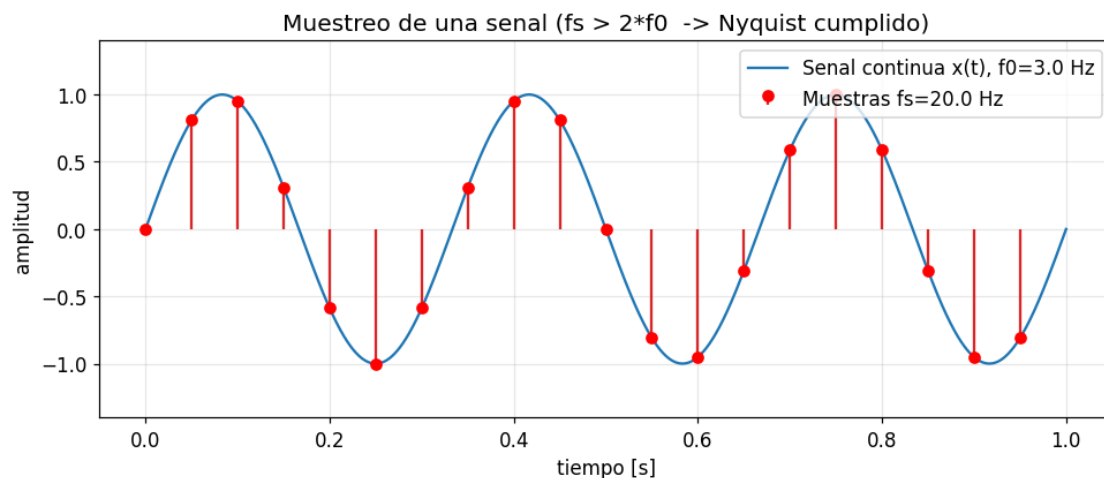
Teorema de muestreo (Nyquist-Shannon): para poder reconstruir sin pérdida una señal de banda limitada con frecuencia máxima f_{max} , hay que muestrear a

$$f_s > 2 f_{max}$$

La frontera $f_N = f_s/2$ es la **frecuencia de Nyquist**: el ADC solo “ve” sin ambigüedad frecuencias por debajo de ella.

Ejemplo numérico: audio con contenido hasta $f_{max} = 20 \text{ kHz} \Rightarrow f_s > 40 \text{ kHz}$. Por eso el CD usa $f_s = 44.1 \text{ kHz}$ (margen para el filtro anti-aliasing real, que no es ideal).

La figura compara una señal “continua” (muy sobremuestreada) con sus muestras a una f_s razonable.



$f_{\text{Nyquist}} = f_s/2 = 10.0 \text{ Hz} > f_0 = 3.0 \text{ Hz} \rightarrow \text{reconstruible}$

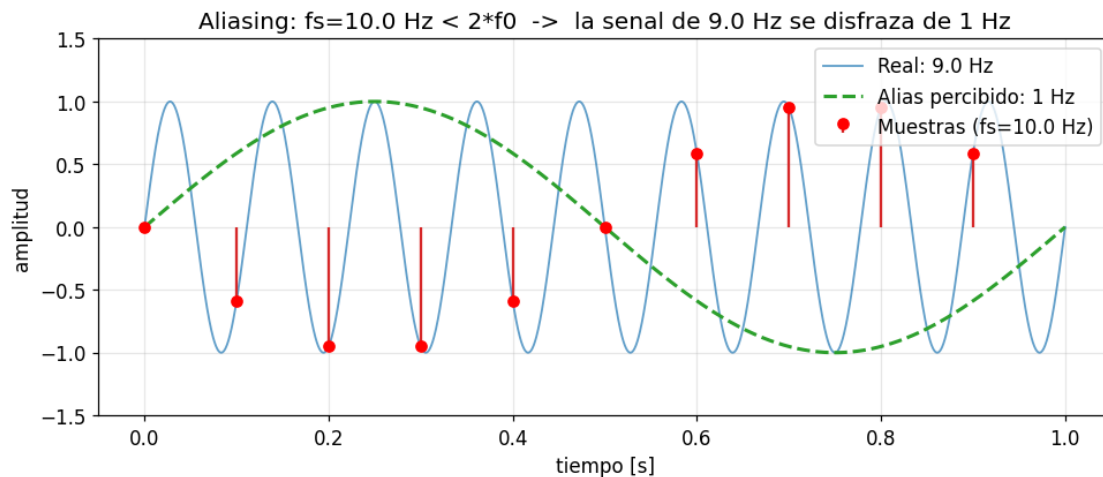
3. Aliasing (solapamiento espectral)

Resultado clave: si $f_s \leq 2f_0$, una senoide de frecuencia f_0 se hace pasar por otra de frecuencia

$$f_{\text{alias}} = |f_0 - k f_s| \quad (\text{el } k \text{ entero que la lleva a } [0, f_s/2])$$

Las muestras de las dos señales **coinciden exactamente**: el ADC no puede distinguirlas. Por eso es imprescindible el **filtro anti-aliasing** antes de muestrear.

Ejemplo: $f_0 = 9 \text{ Hz}$ muestreada a $f_s = 10 \text{ Hz} \Rightarrow f_{\text{alias}} = |9 - 10| = 1 \text{ Hz}$. La figura muestra cómo las muestras de la señal de 9 Hz caen sobre la senoide aliada de 1 Hz.



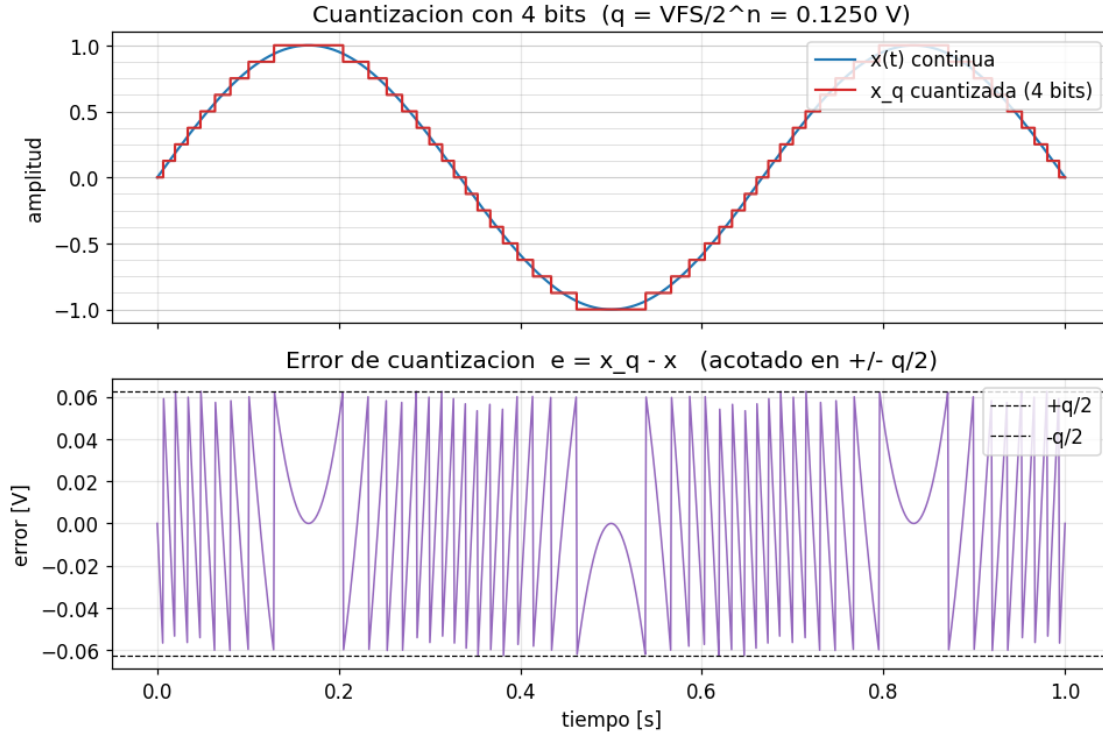
$$f_{\text{alias}} = |9.0 - 1 \cdot 10.0| = 1 \text{ Hz}$$

4. Cuantización y error de cuantización

Tras muestrear (discretizar en tiempo) hay que **cuantizar** (discretizar en amplitud): redondear cada muestra al nivel más cercano de entre 2^n niveles.

Definiciones clave (examen): - **Escalón / LSB:** $q = \frac{V_{FS}}{2^n}$ (rango de fondo de escala V_{FS} dividido entre el n° de niveles). - **Error de cuantización:** $e = x_q - x$, acotado en $-q/2 \leq e \leq +q/2$ (con redondeo). - Modelado como **ruido** uniforme: su valor eficaz es $e_{RMS} = \frac{q}{\sqrt{12}}$.

A más bits $\Rightarrow$ menor $q \Rightarrow$ menor error $\Rightarrow$ mejor resolución. La figura muestra la señal escalonada (cuantizada) y, debajo, el error en “diente de sierra” confinado entre $\pm q/2$.



$q = 0.1250 \text{ V}$ | error max teorico = $\pm q/2 = \pm 0.0625 \text{ V}$ | observado = $\pm 0.0625 \text{ V}$

$e_{\text{RMS}} \text{ teorico} = q/\sqrt{12} = 0.0361 \text{ V}$ | medido = 0.0345 V

5. Relación señal-ruido (SNR) y ENOB

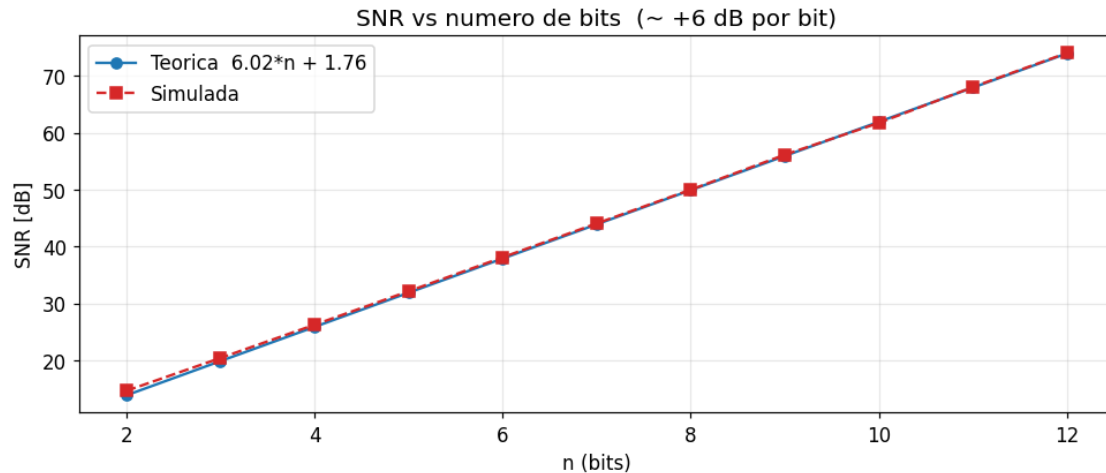
El ruido de cuantización fija un límite de calidad. Para una senoide a fondo de escala con n bits:

$$\text{SNR}_{dB} = 6.02 n + 1.76$$

(cada bit $\sim +6 \text{ dB}$ de SNR). A la inversa, dada la SNR real medida se obtiene el **número efectivo de bits**:

$$\text{ENOB} = \frac{\text{SNR}_{dB} - 1.76}{6.02}$$

El ENOB es $< n$ en convertidores reales por ruido, no linealidad y errores dinámicos. La figura verifica empíricamente la regla de los 6 dB/bit comparando la SNR teórica con la medida por simulación.



n= 2 bits ->	SNR teorica	13.80 dB		simulada	14.59 dB
n= 3 bits ->	SNR teorica	19.82 dB		simulada	20.37 dB
n= 4 bits ->	SNR teorica	25.84 dB		simulada	26.22 dB
n= 5 bits ->	SNR teorica	31.86 dB		simulada	32.12 dB
n= 6 bits ->	SNR teorica	37.88 dB		simulada	38.10 dB
n= 7 bits ->	SNR teorica	43.90 dB		simulada	44.06 dB
n= 8 bits ->	SNR teorica	49.92 dB		simulada	50.01 dB
n= 9 bits ->	SNR teorica	55.94 dB		simulada	56.17 dB
n=10 bits ->	SNR teorica	61.96 dB		simulada	61.76 dB
n=11 bits ->	SNR teorica	67.98 dB		simulada	68.09 dB
n=12 bits ->	SNR teorica	74.00 dB		simulada	74.09 dB

6. DAC: idea, peso binario y red R-2R

Un DAC genera $V_{out} = V_{ref} D/2^n$. Dos arquitecturas clásicas:

1.1.1 6.1 DAC de pesos binarios (suma ponderada)

Cada bit conmuta una corriente/tensión proporcional a su peso usando resistencias en progresión binaria ($R, 2R, 4R, 8R, \dots$ o, como en los apuntes, $R_1 = 2^4 R$, $R_2 = 2^3 R, \dots$). Un amplificador suma las contribuciones.

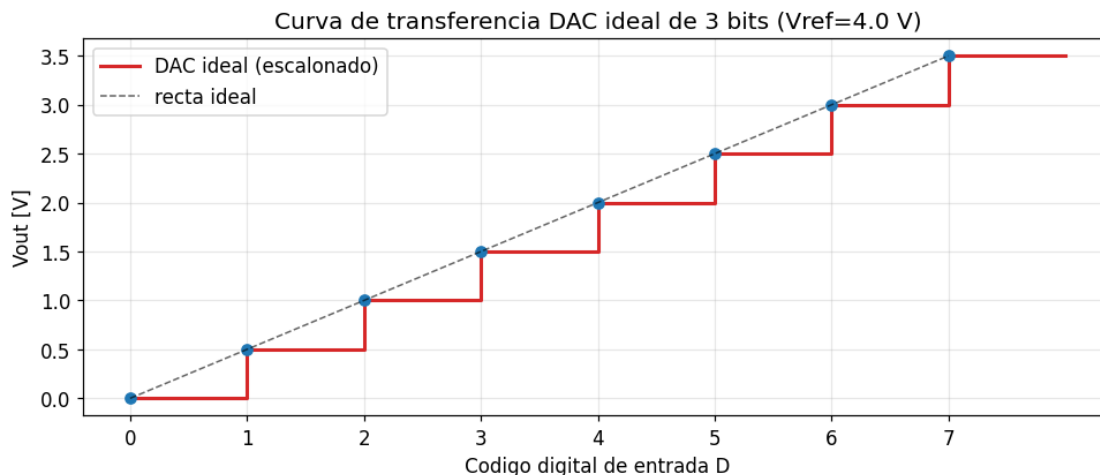
- (+) Conceptualmente simple.
- (-) Necesita resistencias **muy dispares** (para 12 bits, relación 1:2048) -> difícil de fabricar con precisión.

1.1.2 6.2 DAC en escalera R-2R

Usa **solo dos valores** de resistencia: R y $2R$. La red divide la corriente a la mitad en cada nodo, generando automáticamente los pesos binarios.

- (+) Solo dos valores de R (fácil de igualar/ajustar en integrado), pocos interruptores (uno por bit).
- (-) Algo más lenta (las conmutaciones tienen capacidades parásitas).

La fórmula de salida es la misma. La figura muestra la **curva de transferencia escalonada** de un DAC ideal de 3 bits: a cada código de entrada (0-7) le corresponde un escalón de tensión.



$$\text{LSB} = V_{\text{ref}}/2^n = 0.5 \text{ V}$$

D=000 (0) → Vout = 0.000 V

D=001 (1) → Vout = 0.500 V

D=010 (2) → Vout = 1.000 V

D=011 (3) → Vout = 1.500 V

D=100 (4) → Vout = 2.000 V

D=101 (5) → Vout = 2.500 V

D=110 (6) → Vout = 3.000 V

D=111 (7) → Vout = 3.500 V

Verificación numérica de la red R-2R (modelo de corrientes): cada bit aporta una corriente que se reparte por la escalera, y la suma reproduce exactamente $V_{\text{ref}} D/2^n$. Lo comprobamos para todos los códigos de 4 bits.

R-2R reproduce la formula ideal para los 16 codigos: True

Ej. D=1011 (11): 3.4375 V (= 3.4375 V)

7. Curvas de transferencia y errores estáticos

La curva real se desvía de la ideal. Errores **estáticos**:

Error	Tipo	Qué provoca
Offset	lineal	desplaza toda la curva (la salida con código 0 no es 0).
Ganancia	lineal	cambia la pendiente; Ganancia = FS/V_{ref} ; <i>error de ganancia</i> = pendiente real != ideal.

Error	Tipo	Qué provoca
DNL (No-Linealidad Diferencial)	no lineal	un escalón mide más/menos de 1 LSB. DNL ≤ -1 LSB $\Rightarrow$ código ausente (missing code).
INL (No-Linealidad Integral)	no lineal	desviación acumulada respecto a la recta ideal; INL = suma de DNLs .

Monotonicidad: si $|DNL| < 1$ LSB en todos los códigos, el convertidor es **monótono**. Conocer INL y DNL permite afirmarlo (responde al ejercicio de los apuntes).

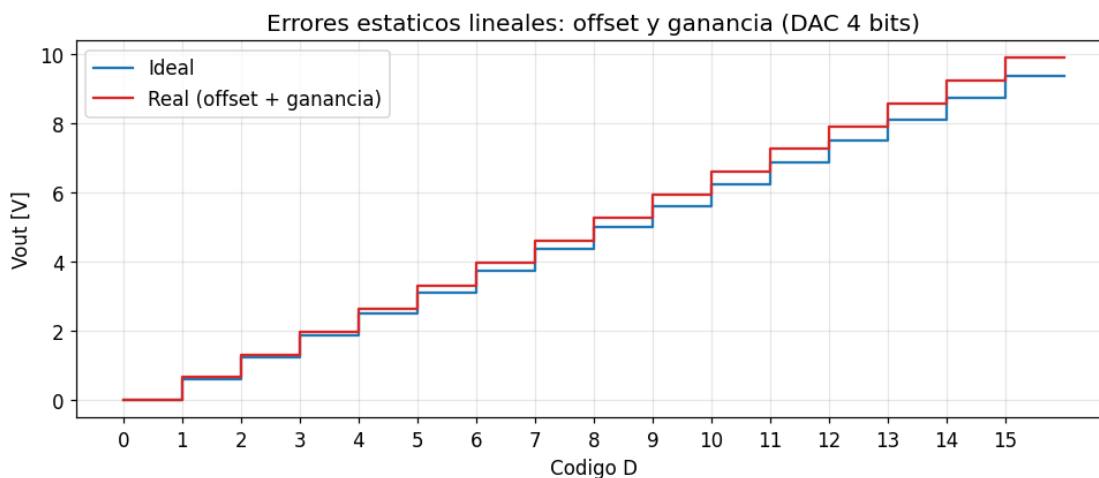
Worked example (ejercicio del Tema 11): DAC de 4 bits, $V_{ref} = 10\text{ V}$, ideal $0000 \rightarrow 0\text{ V}$. Real: $0000 \rightarrow 6\text{ mV}$ y $1111 \rightarrow 9.9\text{ V}$. Calcular ganancia real, error de ganancia y error de offset.

Ganancia ideal = 625.000 mV/código (= LSB ideal)

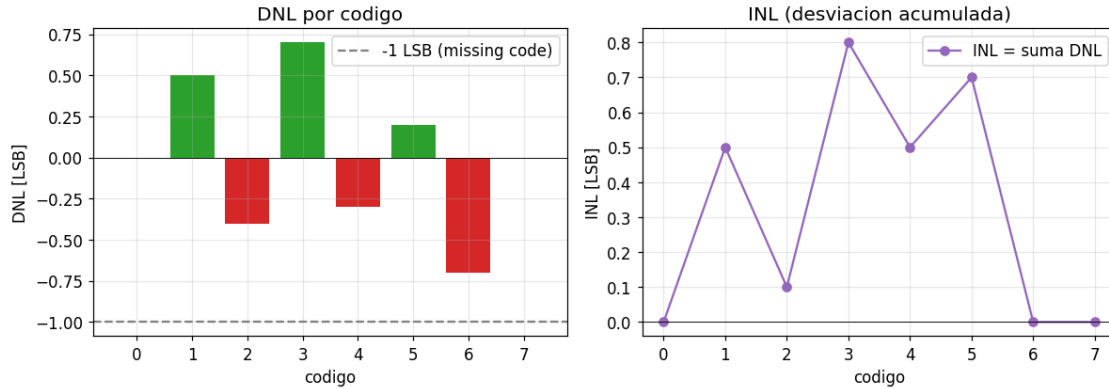
Ganancia real = 659.600 mV/código

Error de offset = 6.0 mV

Error de ganancia = 34.600 mV/código (+5.54 %)



La siguiente figura ilustra **DNL e INL** con un ADC al que le inyectamos deliberadamente escalones de tamaño irregular: se ve cómo el INL acumula esas desviaciones (INL = suma de DNL).



DNL: [0. 0.5 -0.4 0.7 -0.3 0.2 -0.7 0.]

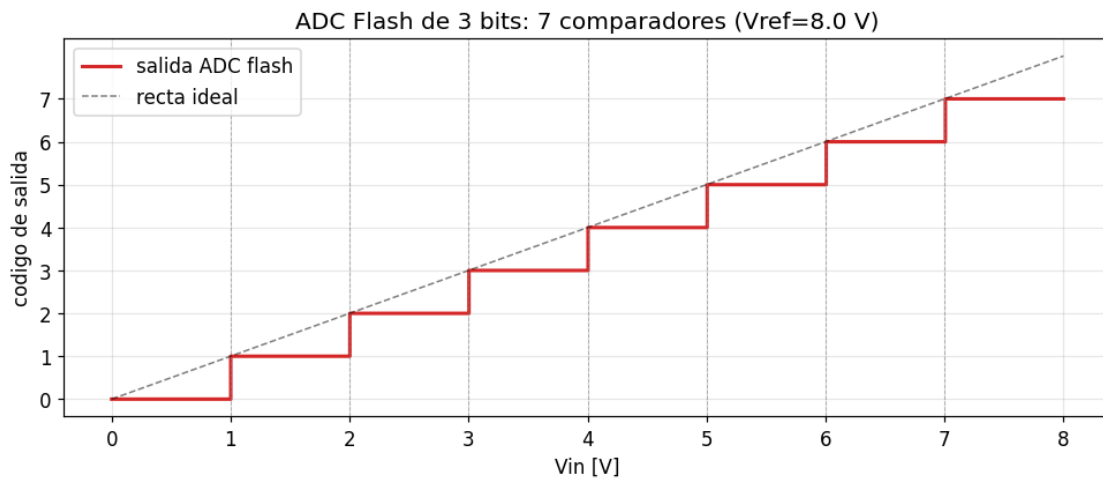
INL: [0. 0.5 0.1 0.8 0.5 0.7 0. 0.] (INL = suma acumulada de DNL)

8. ADC Flash (paralelo)

El más rápido. Compara la entrada simultáneamente contra $2^n - 1$ niveles de referencia mediante $2^n - 1$ comparadores; un codificador con prioridad genera el binario.

- (+) **Una sola conversión** (latencia mínima) -> ideal para muy alta velocidad (RF, vídeo).
- (-) $2^n - 1$ **comparadores** => coste/área explotan: solo para pocos bits (típ. ≤ 8). Etapa inicial de ADCs mixtos.
- Para 3 bits -> 7 comparadores (no hace falta el del cero). El código binario lo fija el comparador activo de mayor orden.

La figura muestra la **escala de cuantización** del flash: las bandas de entrada y el código que entrega.



Umbral de los comparadores [V]: [1. 2. 3. 4. 5. 6. 7.]

9. ADC de aproximaciones sucesivas (SAR)

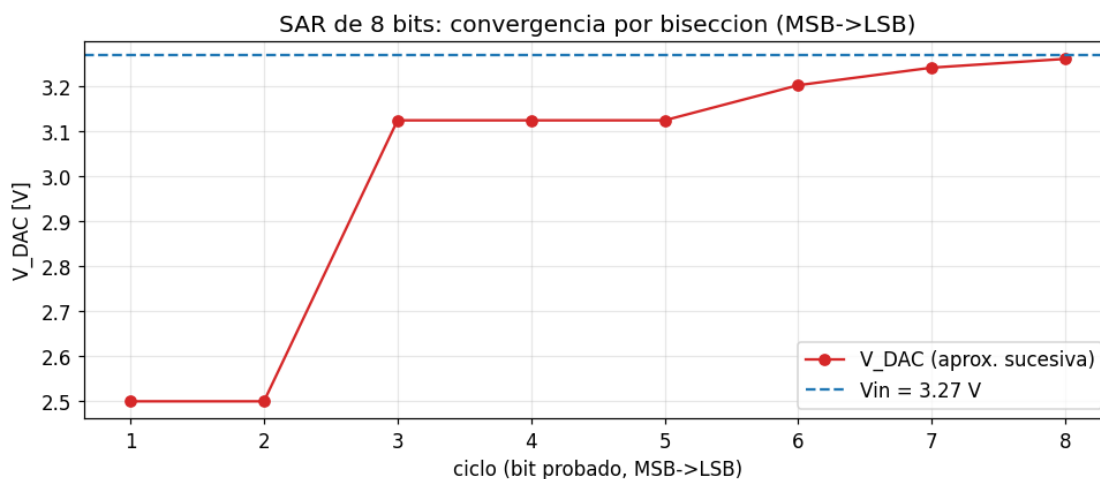
El “caballo de batalla” industrial (8-16 bits). Es de **bucle cerrado**: tiene un DAC interno, compara y decide bit a bit, desde el **MSB** al **LSB** — como adivinar un número por bisección.

Algoritmo ($n+1$ ciclos de reloj, **tiempo de conversión constante** $\sim n$): 1. Pon a prueba el bit más alto a 1; el DAC genera esa tensión. 2. Si $V_{in} \geq V_{DAC} \Rightarrow$ el bit se queda en 1; si no $\Rightarrow$ a 0. 3. Repite con el siguiente bit. Cada paso **divide el intervalo por 2**.

La figura traza la tensión del DAC interno aproximándose a la entrada en cada ciclo (búsqueda binaria).

```
bit 7: pruebo VDAC=2.5000 V -> 1 | codigo=10000000
bit 6: pruebo VDAC=3.7500 V -> 0 | codigo=10000000
bit 5: pruebo VDAC=3.1250 V -> 1 | codigo=10100000
bit 4: pruebo VDAC=3.4375 V -> 0 | codigo=10100000
bit 3: pruebo VDAC=3.2812 V -> 0 | codigo=10100000
bit 2: pruebo VDAC=3.2031 V -> 1 | codigo=10100100
bit 1: pruebo VDAC=3.2422 V -> 1 | codigo=10100110
bit 0: pruebo VDAC=3.2617 V -> 1 | codigo=10100111
```

$V_{in}=3.27$ V -> codigo 10100111 (167) -> 3.2617 V



10. ADC de rampa (simple y doble pendiente)

1.1.3 10.1 Rampa simple

Una rampa $V_C = \frac{V_{ref}}{RC}t$ sube mientras un contador cuenta. Cuando la rampa alcanza V_{in} , se para: el valor del contador es proporcional a V_{in} .

$$T_c = RC \frac{V_{in}}{V_{ref}}, \quad D = f T_c = \frac{f V_{in}}{RC V_{ref}}$$

- (+) Simple, barato. (-) **Depende de R, C y f** -> se descalibra con la deriva de componentes.

1.1.4 10.2 Rampa doble (doble pendiente) — el truco que cancela R, C

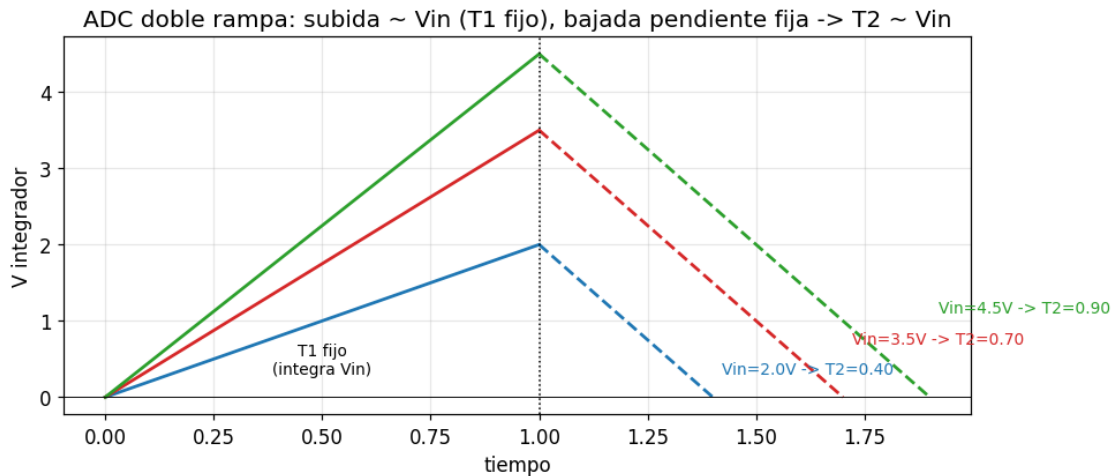
1. **Integra V_{in} un tiempo fijo $T_1 = 2^n/f$** (cuenta hasta desbordar): la carga depende de V_{in} .
2. **Descarga con $-V_{ref}$** contando hasta pasar por cero, en tiempo T_2 .

Como $V_{in}T_1 = V_{ref}T_2$, resulta:

$$D = \frac{V_{in}}{V_{ref}} 2^n$$

R, C y f se cancelan -> muy preciso y robusto (lo usan los multímetros). (-) Lento.

La figura muestra las dos pendientes: la subida depende de V_{in} , la bajada tiene pendiente fija; T_2 codifica el resultado.

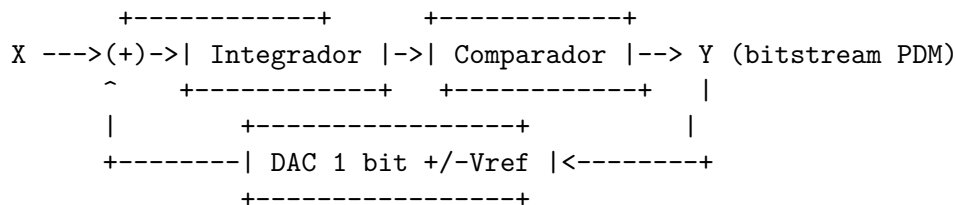


$D = (V_{in}/V_{ref}) * 2^n$ -> R, C y f se cancelan (independiente de la deriva)

11. Convertidores sobremuestreados: Sigma-Delta (Sigma-Delta)

Idea central: cambiar **resolución por velocidad**. Se usa un convertidor de **1 bit** pero a frecuencia 2^n veces mayor que Nyquist; por cada bit que “quitas” de resolución, **duplicas la frecuencia**. La salida es un **bitstream** (densidad de unos proporcional al valor de la señal), que luego un filtro pasa-bajo promedia.

Modulador Sigma-Delta de 1.er orden: un integrador + un comparador (cuantizador de 1 bit) + realimentación con un DAC de 1 bit que devuelve $\pm V_{ref}$:



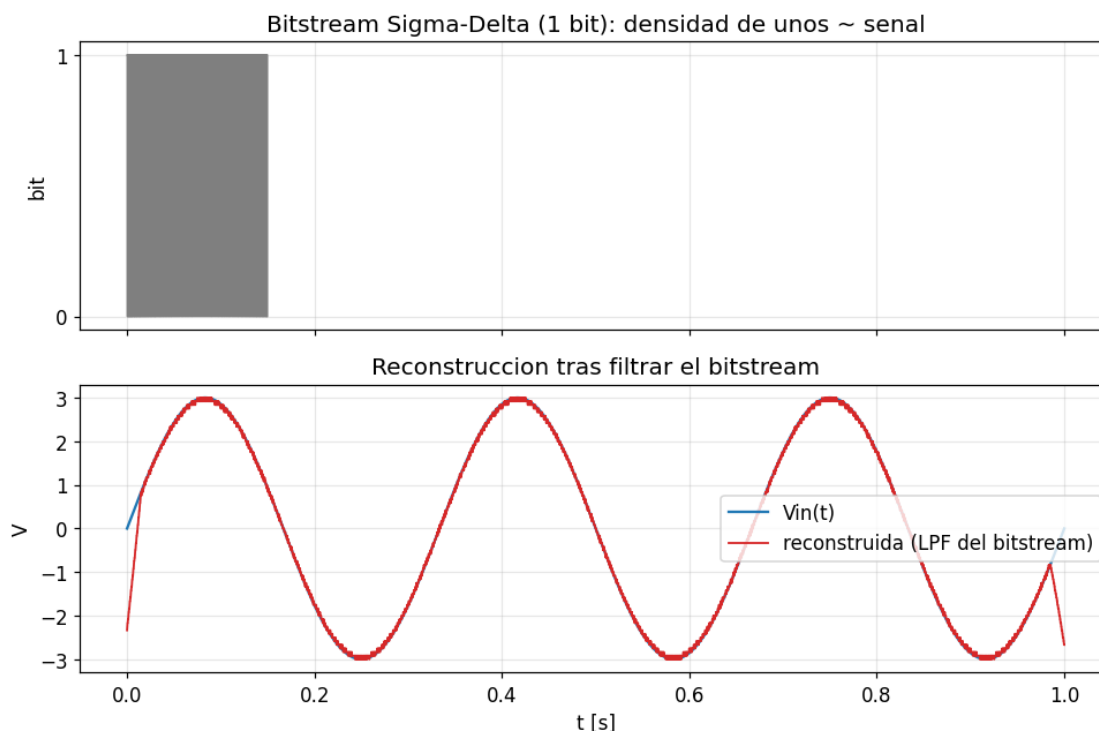
El lazo **fuerza que la media del bitstream siga a la entrada**. Empuja el ruido de cuantización hacia altas frecuencias (*noise shaping*), donde el filtro lo elimina -> gran SNR en banda base.

Worked example (Tema 12.2): $V_{ref} = \pm 5$, $V_{in} = 2V$. Reproducimos el modulador en código y obtenemos el bitstream:

Bitstream (8): 10111011

Densidad de unos = $6/8 = 0.750$

Tension reconstruida $\sim 2.500V$ ($V_{in} = 2.0V$)



Resumen Sigma-Delta (examen): - Para n bits $\Rightarrow 2^n$ operaciones, pero **muy sencillas** (1 bit cada una). - La salida hay que **filtrarla** (LPF) para obtener el valor. - Más complejo de *entender*, más simple de *implementar* (poca parte analógica, casi todo digital). - Mejora la SNR por *noise shaping*. Variantes D/A: **PWM** (una conmutación por ciclo, espectro concentrado) y **PDM** (secuencia pseudoaleatoria, espectro disperso -> menos tono de conmutación).

12. CAD: flujo de diseño digital (Computer-Aided Design)

El diseño de circuitos digitales modernos se hace con herramientas **CAD/EDA**. El flujo top-down típico:

```

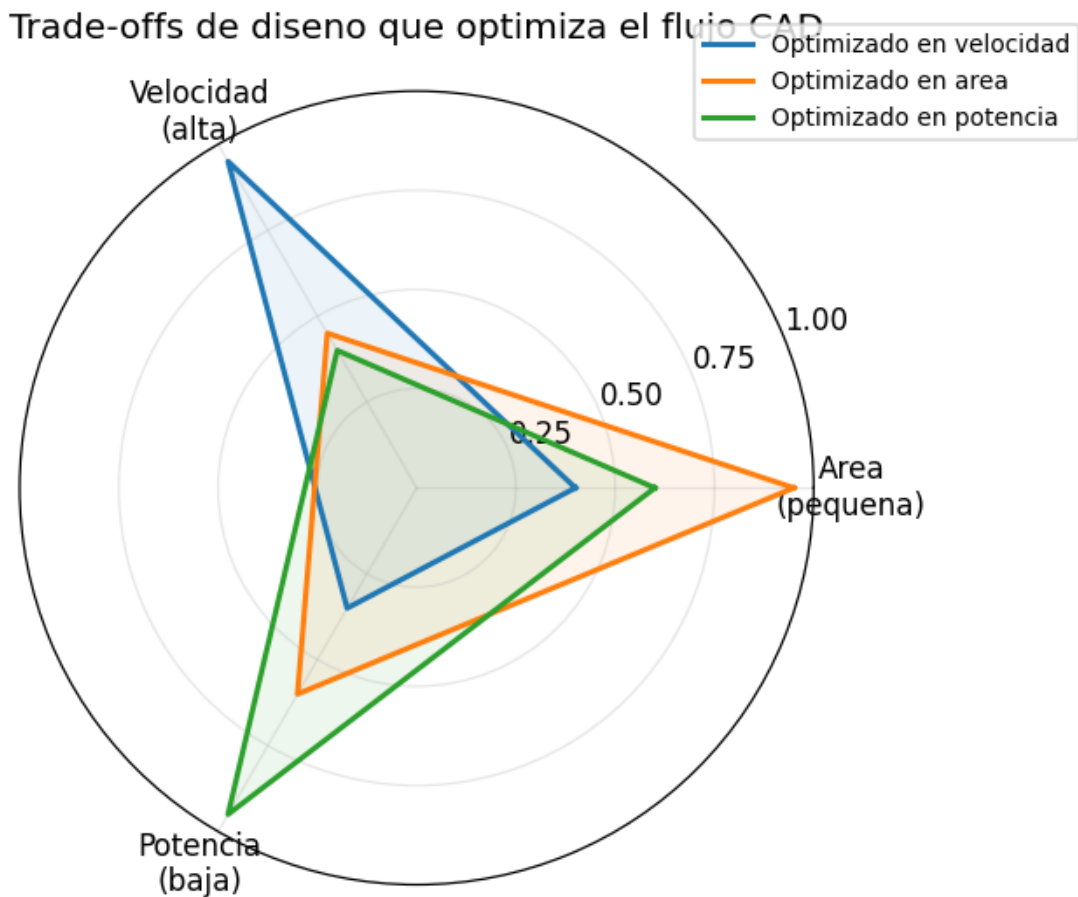
Especificacion
|
v
Descripcion RTL ----- HDL: VHDL / Verilog (comportamiento, registros)
|
v

```

Síntesis lógica ----- -> netlist de puertas + restricciones (timing, área, potencia)
 |
 v
 Verificación ----- simulación (testbench), comprobación funcional/temporal
 |
 v
 Place & Route ----- ubicación de celdas + interconexión (ASIC) / mapeo a LUTs (FPGA)
 |
 v
 Implementación ----- ASIC (máscaras) / FPGA (bitstream de configuración)

Conceptos clave: - **Abstracción por niveles:** comportamiento -> RTL -> puertas -> físico. Cada nivel se verifica antes de bajar. - **HDL:** se *describe* el hardware, no se programa secuencialmente; la síntesis lo traduce a lógica real. - **Restricciones (constraints):** reloj, *setup/hold*, área, potencia — guían a la herramienta. - **ASIC vs FPGA:** ASIC = óptimo pero caro y no reconfigurable; FPGA = reconfigurable, ideal para prototipos. - **Trade-offs** que el CAD optimiza: **área <-> velocidad <-> potencia** (no se pueden maximizar las tres a la vez).

La figura ilustra esos compromisos de diseño de forma cualitativa.



No se pueden maximizar area, velocidad y potencia a la vez: el CAD busca el equilibrio.

13. Resumen y chuleta de examen

Fórmulas que hay que tener memorizadas:

Concepto	Fórmula
Relación DAC	$V_{out} = V_{ref} D/2^n$
Escalón / LSB	$q = V_{FS}/2^n$
Nyquist	$f_s > 2f_{max}, f_N = f_s/2$
Alias	$f_{alias} = f_0 - kf_s $
Error cuantización	$ e \leq q/2, e_{RMS} = q/\sqrt{12}$
SNR ideal	$SNR_{dB} = 6.02 n + 1.76$
ENOB	$(SNR_{dB} - 1.76)/6.02$
Rampa simple	$D = f V_{in}/(RC V_{ref})$
Doble rampa	$D = (V_{in}/V_{ref}) 2^n$ (cancela R,C,f)
INL	$\sum DNL$

Elección de ADC (decisión típica de examen):

ADC	Velocidad	Resolución	Coste/complejidad	Uso típico
Flash	máxima (1 ciclo)	baja (≤ 8 bits)	alto (2^{n-1} comp.)	RF, vídeo, etapa de pipeline
SAR	media (n ciclos, fijo)	media-alta (8-16)	medio	adquisición industrial general
Rampa simple	baja	media	bajo	económico, descalibrable
Doble rampa	muy baja	alta	medio	multímetros (precisión)
Sigma-Delta	baja-media (sobremuestreo)	muy alta (16-24)	bajo analógico, alto digital	audio , sensores de precisión

Ideas transversales: - Bucle abierto (flash, rampa simple/doble) = más sencillo y/o rápido pero menos preciso o calibrable. - Bucle cerrado (rampa digital, seguimiento, SAR) = DAC interno + comparación + decisión = más preciso. - Todo ADC necesita **S&H**; antes, un **filtro anti-aliasing** obligatorio. - Sigma-Delta y CAD comparten filosofía: **trasladar complejidad de lo analógico a lo digital**, que escala mejor.

Notebook de apuntes propios — Electrónica Digital, 2.º GIERM Mecatrónica. Generado a partir de los temas oficiales 11, 12 y 12.2.