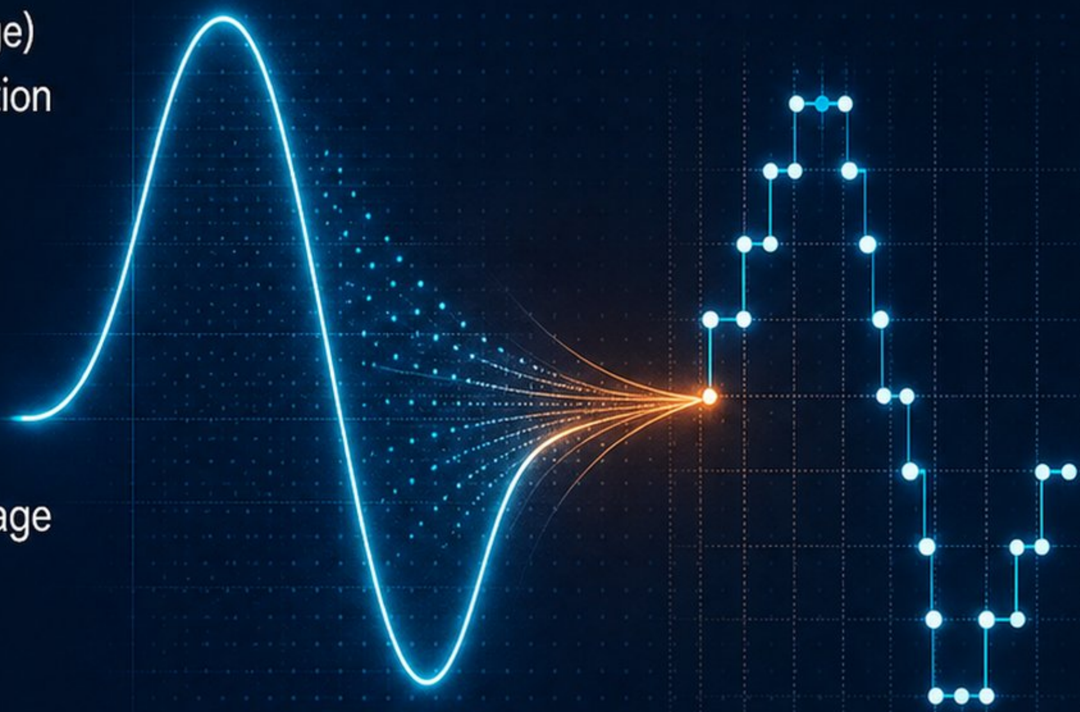
Introduction to Digital Signal Processing

- Signal: a physical quantity (sound, light, voltage) varying with time or space that carries information
- Analog: continuous in time/amplitude (e.g., microphone voltage); Digital: discrete number sequence (e.g., WAV file)
- Why digital? Noise immunity, perfect copies, flexible software algorithms, and compact storage
- DSP: the science of analyzing, modifying, and synthesizing signals in discrete numerical form
- Full chain: analog input → anti-aliasing filter → ADC → processor → DAC → reconstruction filter → analog output

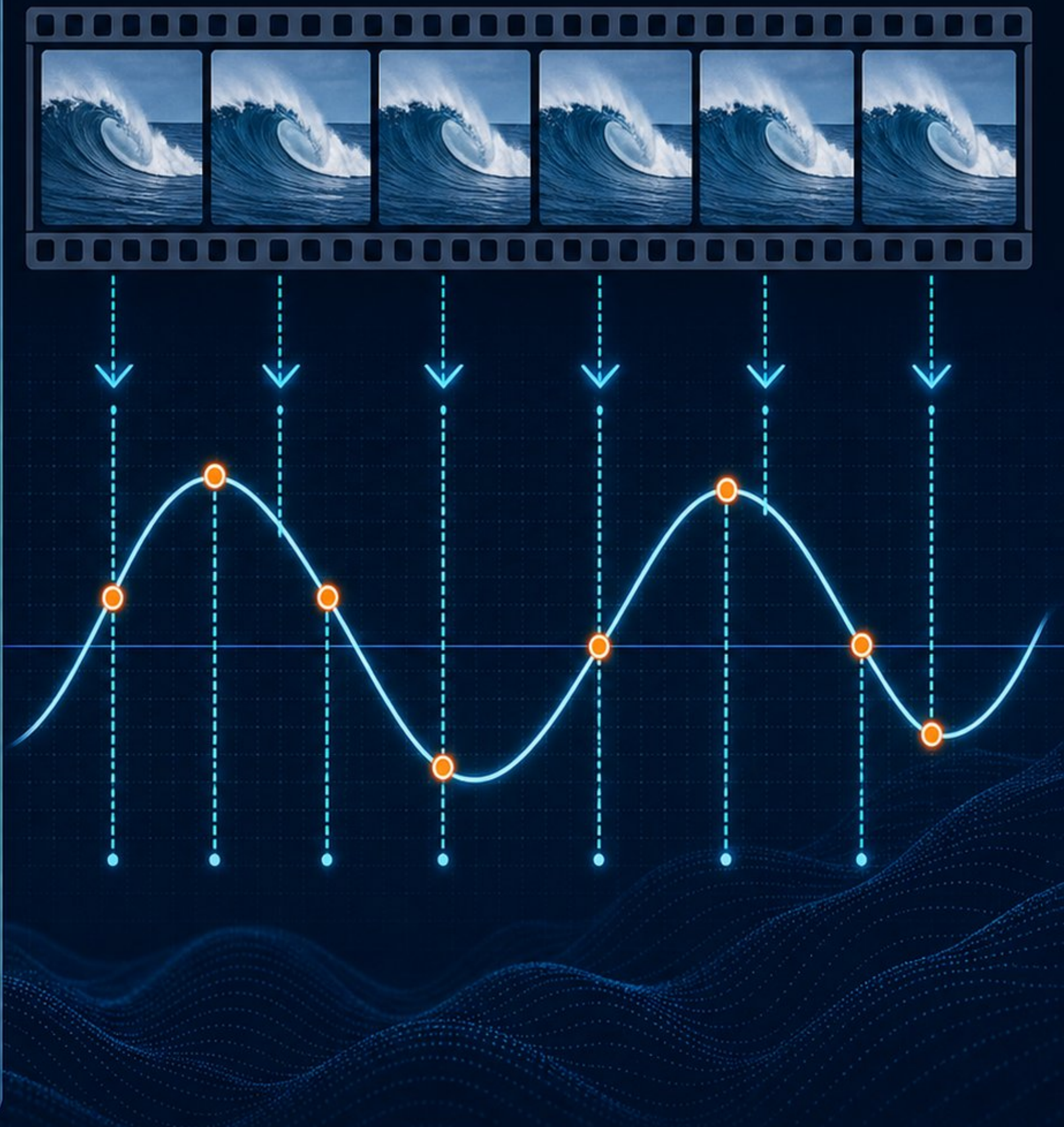


Why Go Digital? Everyday Magic Powered by DSP

- Streaming music, noise-canceling headphones, and voice assistants rely on DSP
- Digital files copy infinitely without degradation, unlike analog tapes
- Same hardware becomes guitar pedal, medical monitor, or radar via software
- On-device AI runs keyword spotting and person detection at milliwatt power
- DSP chain powers wireless comms, biomedical imaging, and audio codecs

The First Step: Sampling — Capturing the World in Snapshots

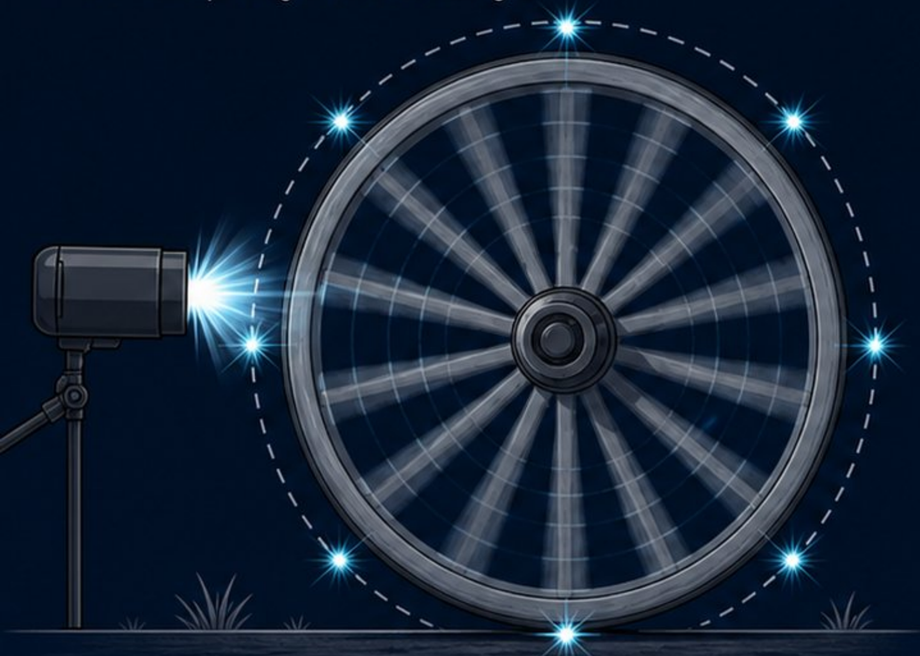
- **Sampling:** measuring a continuous signal's amplitude at regular, discrete instants in time.
- **Camera analogy:** a movie is a series of still frames; an audio sample is a snapshot of air pressure. The sampling rate is the frame rate.
- **Nyquist-Shannon theorem:** to faithfully reconstruct a signal, you must sample at a rate strictly greater than twice the highest frequency present.
- **Nyquist frequency** (half the sampling rate) is the theoretical upper limit for recoverable information.
- **Concrete benchmarks:** CD audio at 44.1 kHz (hearing up to ~20 kHz); telephone speech at 8 kHz (up to ~3.4 kHz); ultrasound imaging at 40–60 MHz.



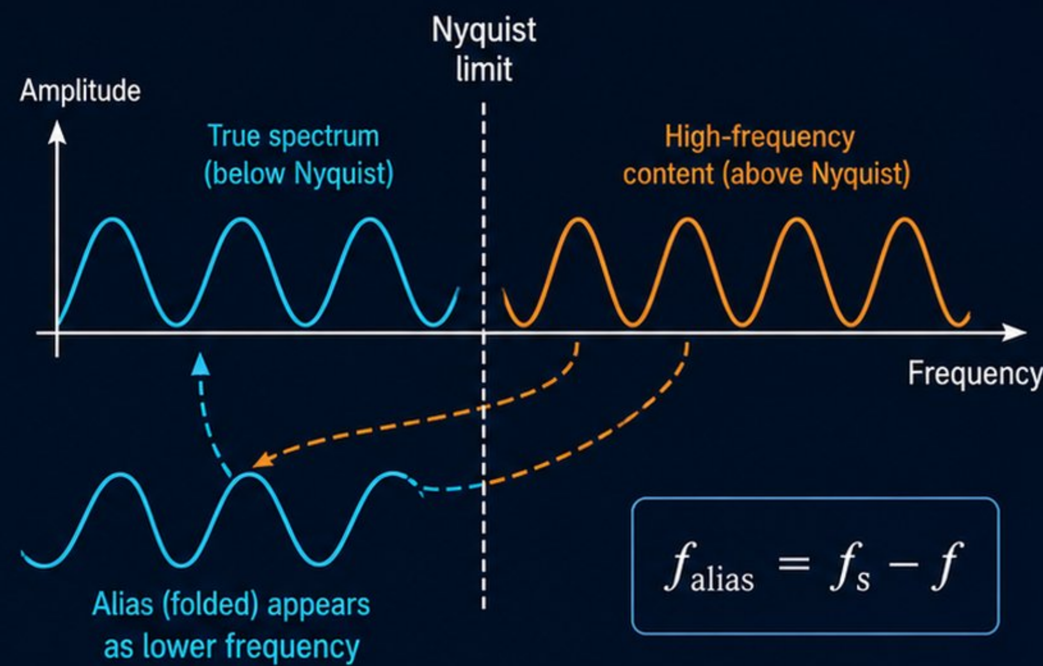
The Specter of Undersampling: Introducing Aliasing

WAGON-WHEEL EFFECT

A wheel spinning under a strobe light



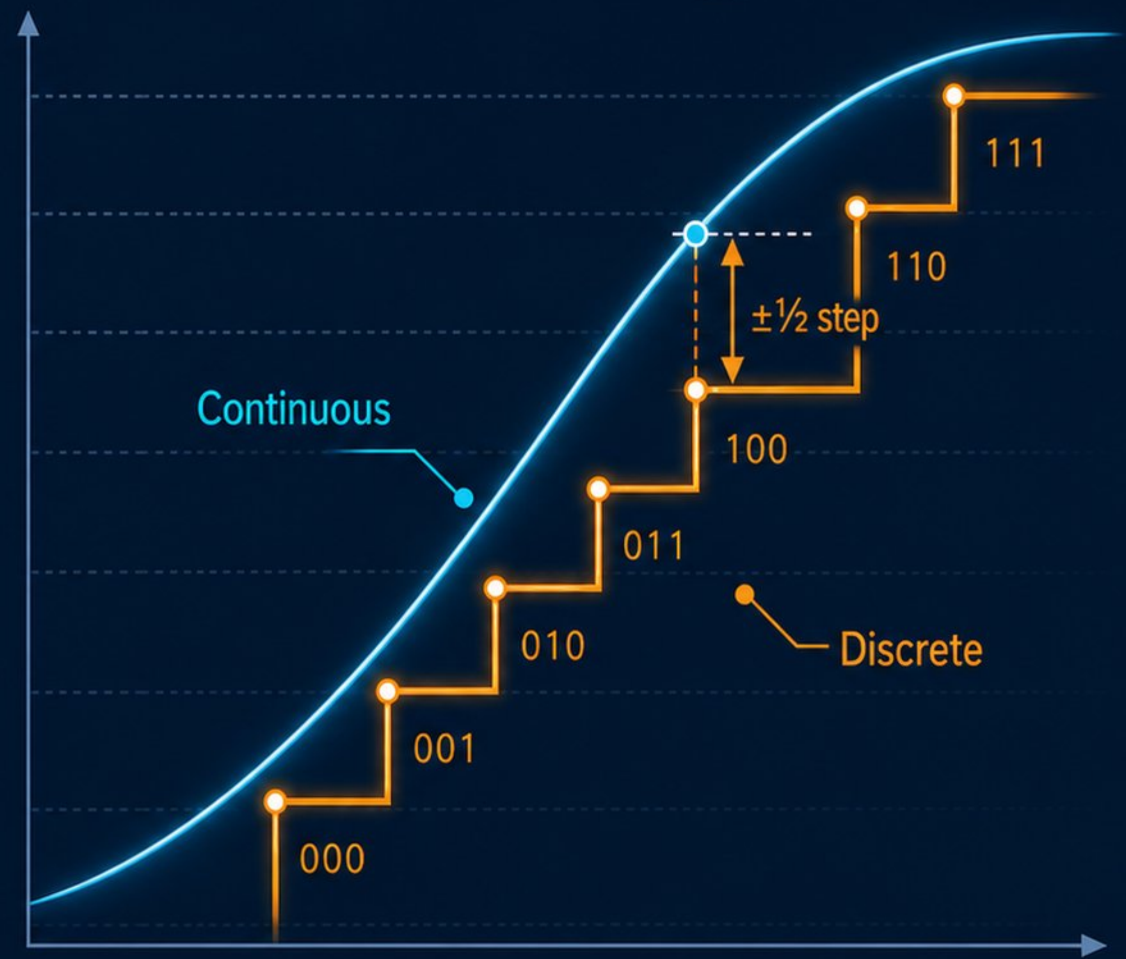
FREQUENCY FOLDING (ALIAS MAPPING)



- **Aliasing:** permanent distortion where high frequencies masquerade as lower false frequencies
- **Wagon-wheel effect:** camera frame rate too low to resolve true wheel rotation
- **Frequency folding:** f above Nyquist maps to alias $f_{\text{alias}} = f_s - f$
- **Example:** 15 kHz tone sampled at 20 kHz creates an irreversible 5 kHz alias
- **Anti-aliasing filter:** analog low-pass filter before ADC removes frequencies above Nyquist limit

Bit by Bit: Quantization and the Staircase of Precision

- Quantization: converting a sample's continuous amplitude to a discrete level from a finite set.
- Bit depth: N-bit gives 2^N levels (8-bit=256, 16-bit=65,536, 24-bit≈16.8M).
- Error bounded by ±½ step; each added bit improves SQNR by roughly 6 dB.
- 8-bit sounds harsh/gritty; 16-bit provides 96 dB dynamic range, transparent for listening.
- Design trade-off: more bits = higher fidelity vs. greater storage and bandwidth.

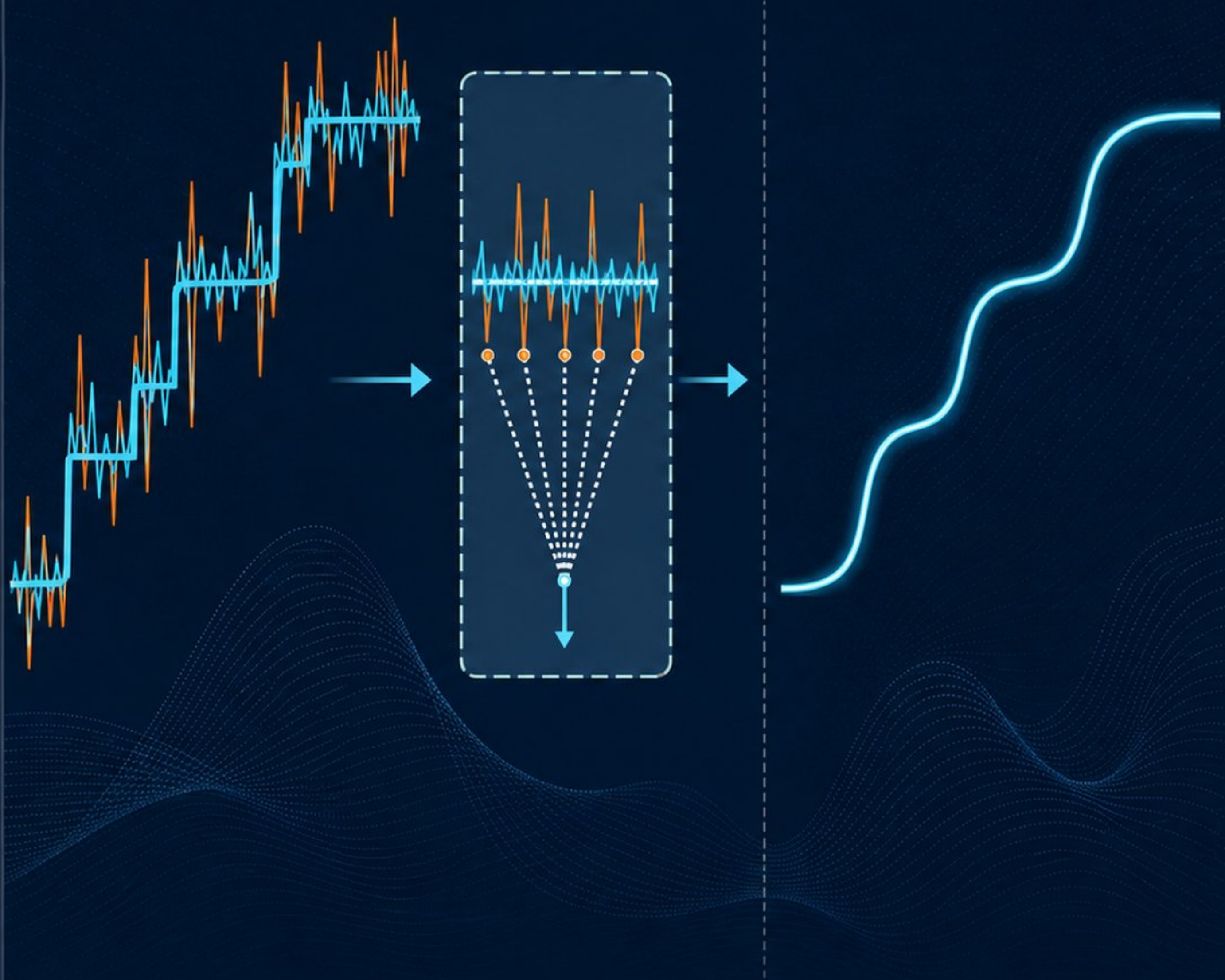


Your Signal as a Number Sequence: Discrete-Time Operations

- Digital signal $x[n]$: integer index n links to physical time via sampling period T
- Shift (delay): $x[n-k]$; Scale (volume): $g \cdot x[n]$; Sum: $x_1[n] + x_2[n]$
- Echo = delayed scaled copy; a mixer blends tracks by summation
- These atoms—delay, scaling, adding—build the molecules of digital filters
- Stem plot shows dots at exact indices, not a smooth line; signal is recoverable with proper sampling

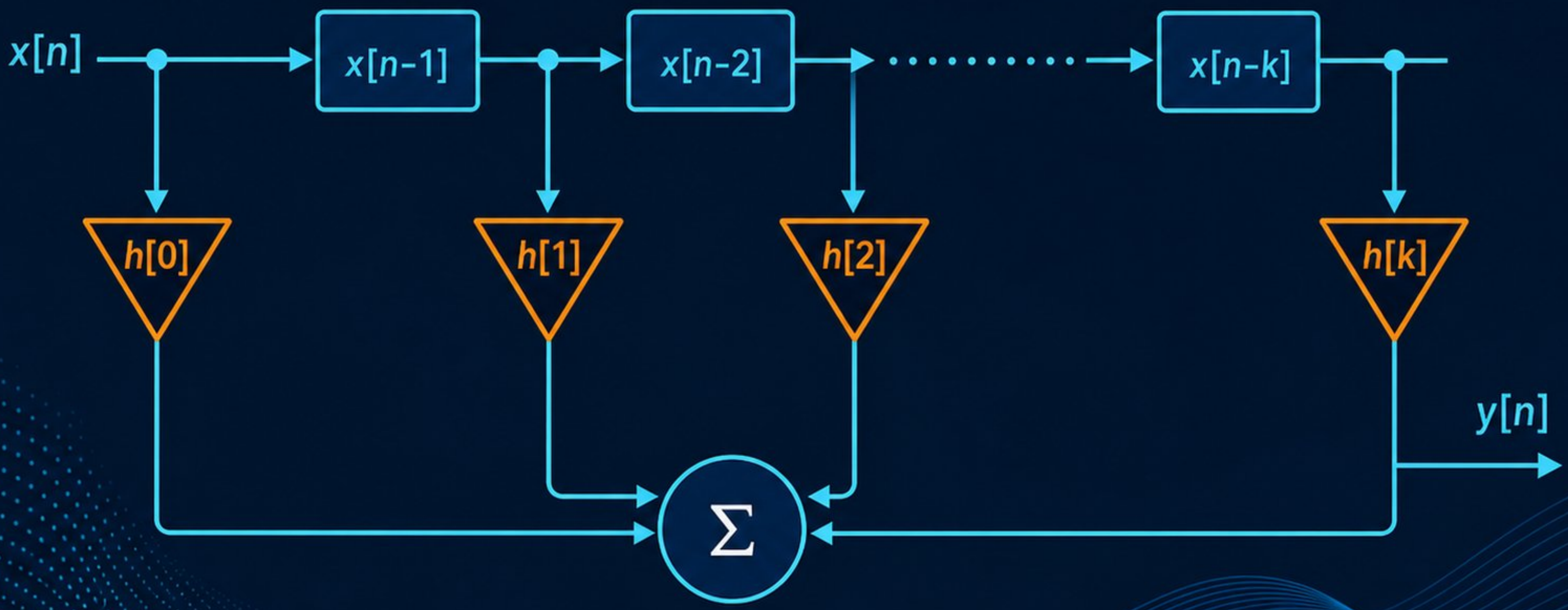
Gentle Smoothing: The Moving-Average Filter

- **Digital filter:** algorithm that selectively modifies signal components
- **Moving-average:** y[n] = average of current + (M-1) past input samples
- **Example:** 5-point average turns noisy temperature staircase into a smooth trend
- **Intuition:** uncorrelated noise cancels out; slowly-changing signal is preserved (crude low-pass)
- **Trade-off:** larger M = smoother output, but rounded edges and delay proportional to window length

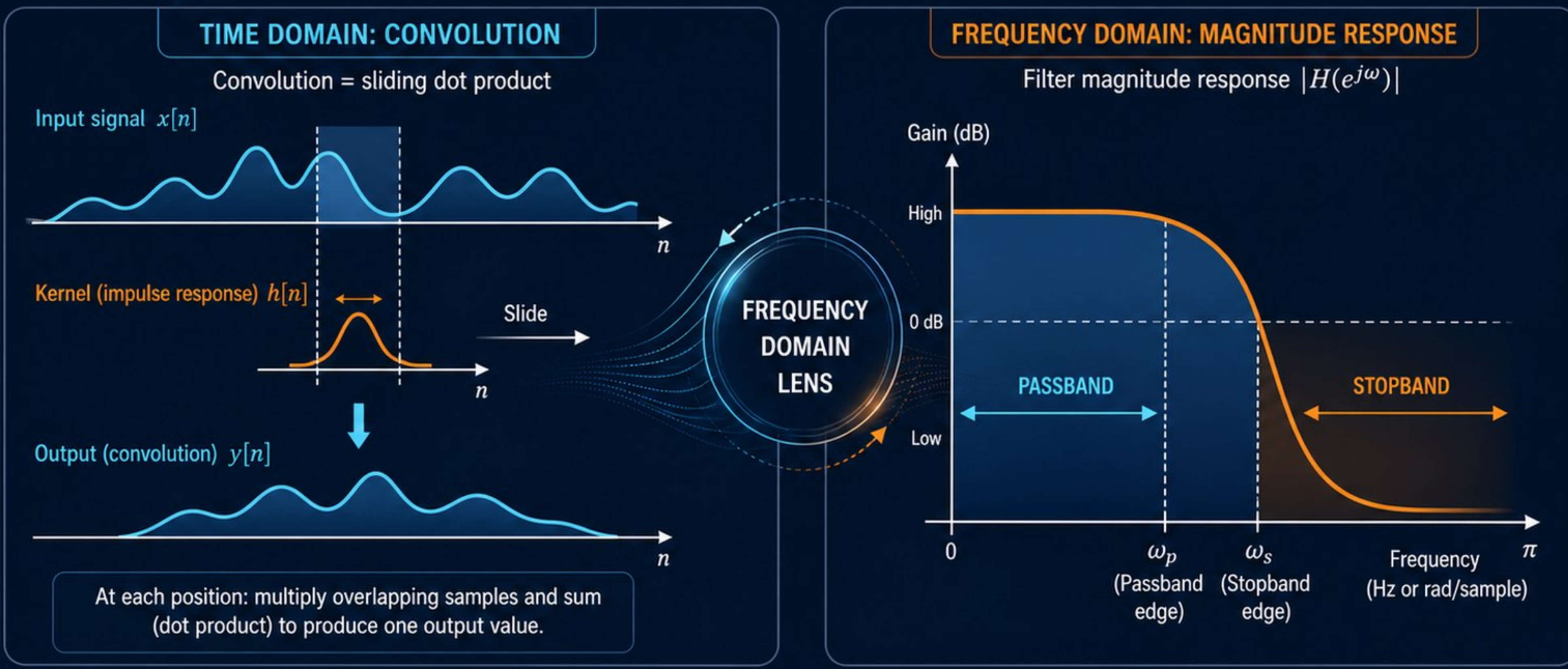


FIR Filters: Weighted Sums of Past and Present

- FIR filter = weighted sum of delayed input samples: $y[n] = \Sigma\ h[k] \cdot x[n-k]$
- Unlike simple moving average, coefficients $h[k]$ can assign different importance to each tap
- Block diagram: delay elements → multipliers → adder — no feedback
- Inherently stable and can achieve exact linear phase (constant group delay)
- Python: scipy.signal.firwin() designs coefficients; lfilter() applies the filter



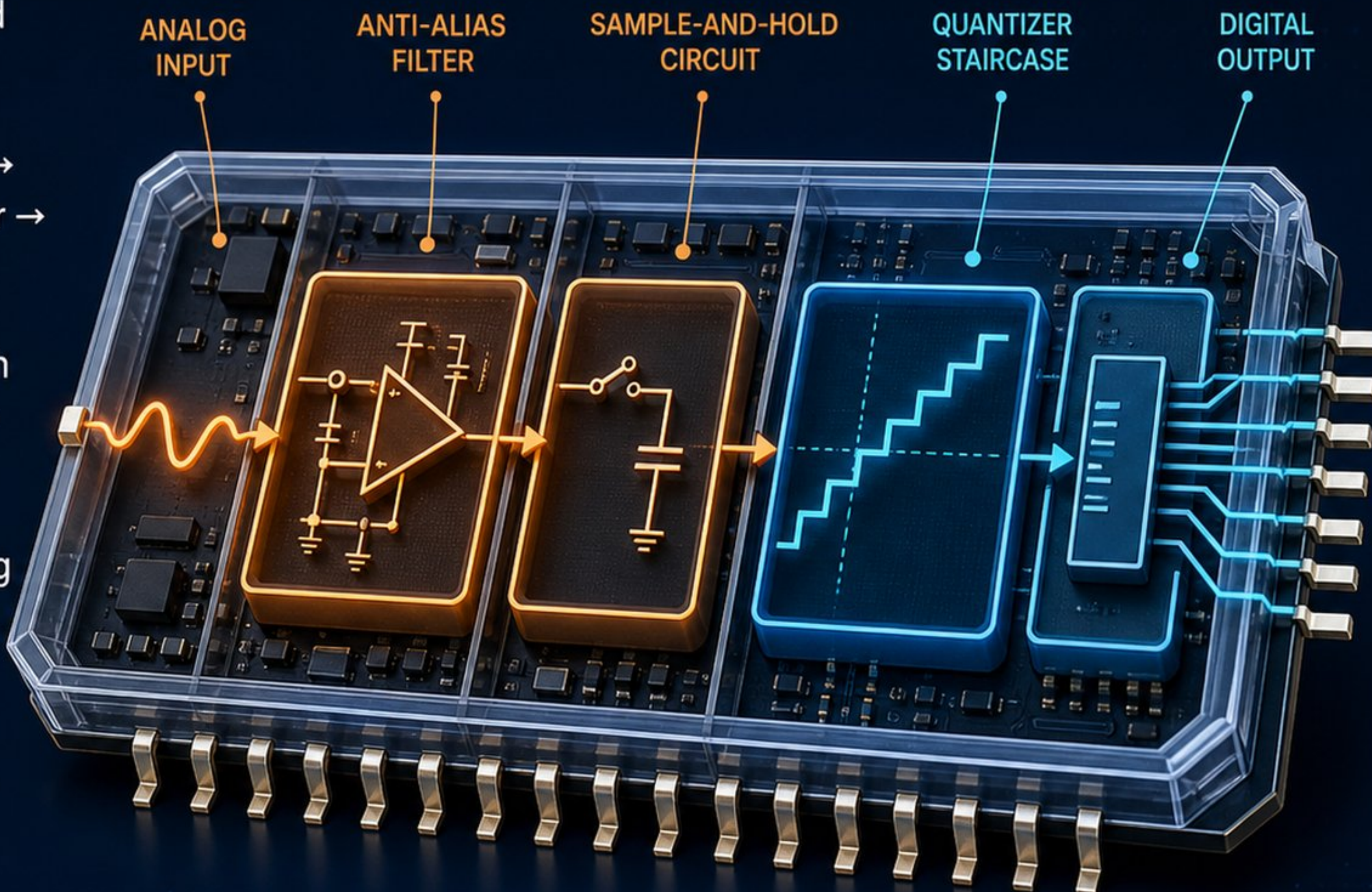
Seeing the Unseen: Time and Frequency Perspectives on Filtering



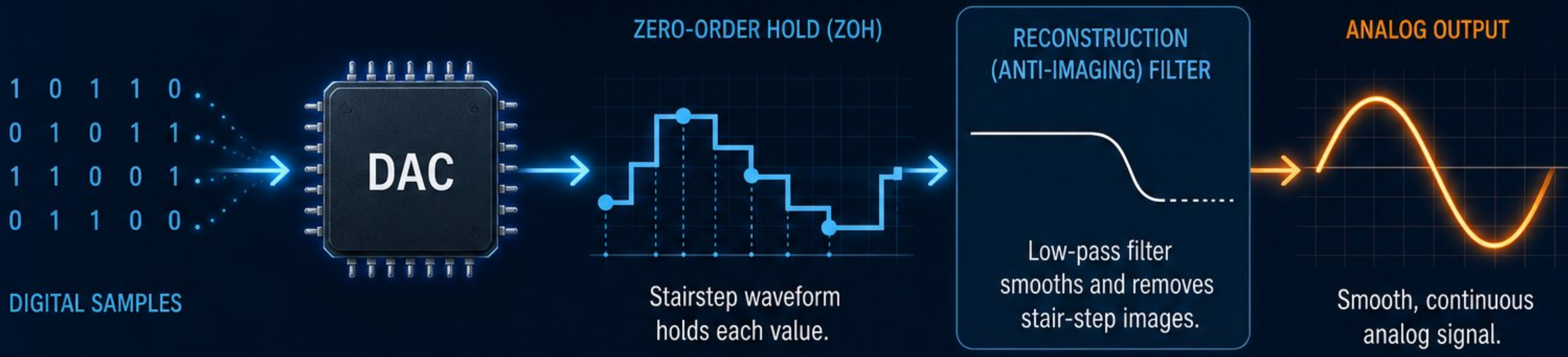
- Frequency domain views amplitude and phase vs. frequency; any signal is a sum of sine waves
- Convolution: sliding dot product of filter's impulse response across the signal
- Amplitude frequency response plot shows gain (dB) vs. frequency (Hz or rad/sample)
- Filter types: low-pass, high-pass, band-pass, band-stop seen clearly in magnitude response
- Wider time-domain smoothing = narrower low-pass band; sharper transitions need more taps

The ADC: Where Analog and Digital Meet

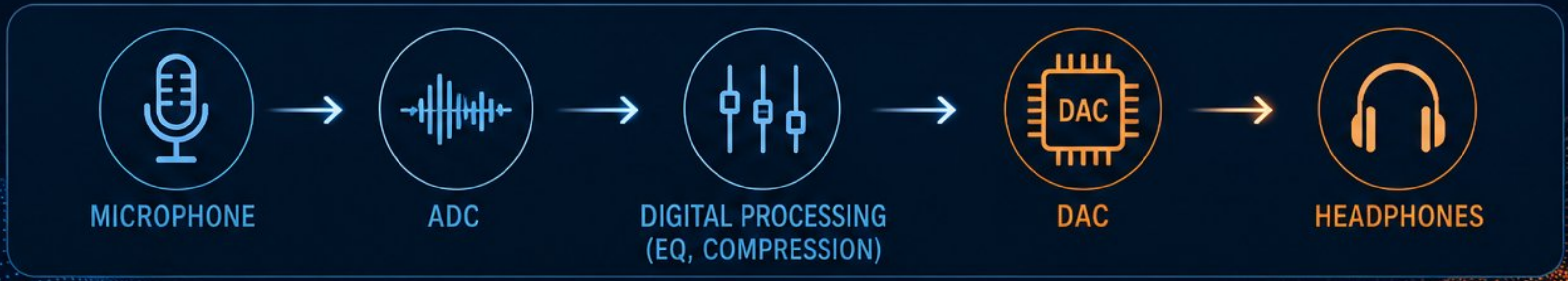
- ADC integrates sampling and quantization in one device
- Core stages: anti-alias filter → sample-and-hold → quantizer → encoder
- Consumer audio ADCs reach 128 dB dynamic range, up to 768 kHz
- 24-bit is professional recording standard; 32-bit emerging for headroom
- Anti-alias filter must precede sampling — the ADC alone cannot prevent aliasing



The DAC: Reconstructing the Analog World

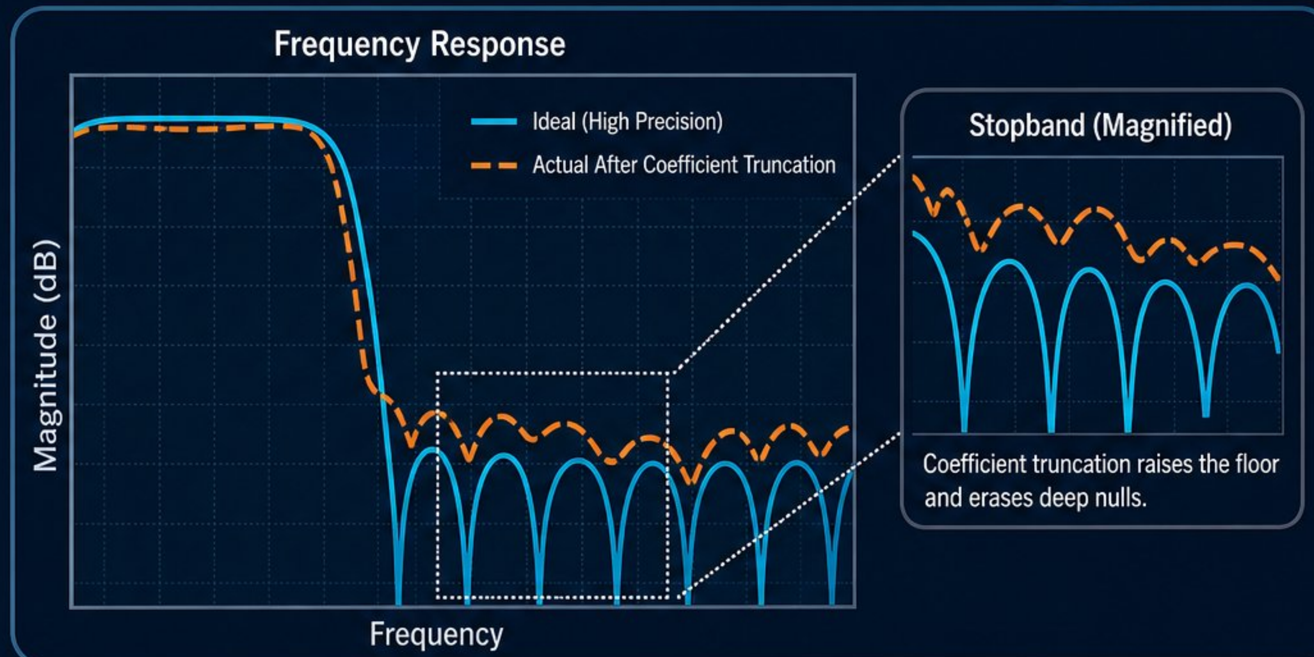


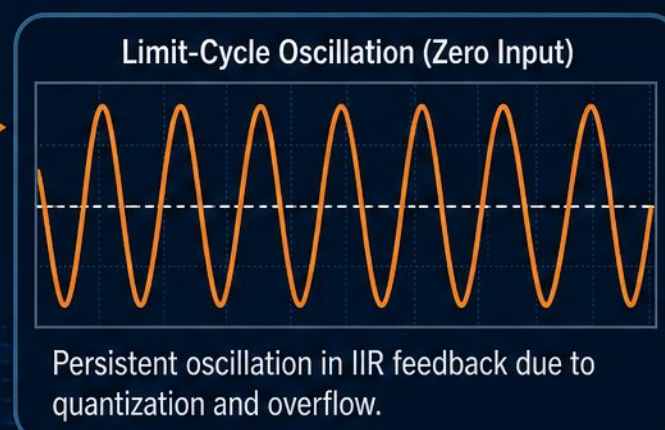
- The DAC converts digital sample values back into a continuous-time, continuous-amplitude voltage signal.
- Zero-order hold (ZOH) reconstruction creates a 'stairstep' waveform by holding each digital value until the next sample.
- The reconstruction (anti-imaging) filter smooths out stair-step images and interpolates a clean analog signal.
- Professional DACs support 32-bit resolution and sample rates to 768 kHz and beyond; DSD uses a different conversion architecture.
- End-to-end chain: microphone → ADC → digital processing (EQ, compression) → DAC → headphones.



Pitfalls in the Real World: Quantization, Overflow, and Instability

- Low-bit quantization (e.g., 4-bit) causes audible, correlated distortion—not just white noise.
- Coefficient rounding in fixed-point can erase deep stopband nulls; re-optimize within the target word length.
- IIR feedback loops risk overflow and persistent limit-cycle oscillations even with zero input.
- Mitigate with biquad cascades, saturation arithmetic, and floating-point when hardware permits.



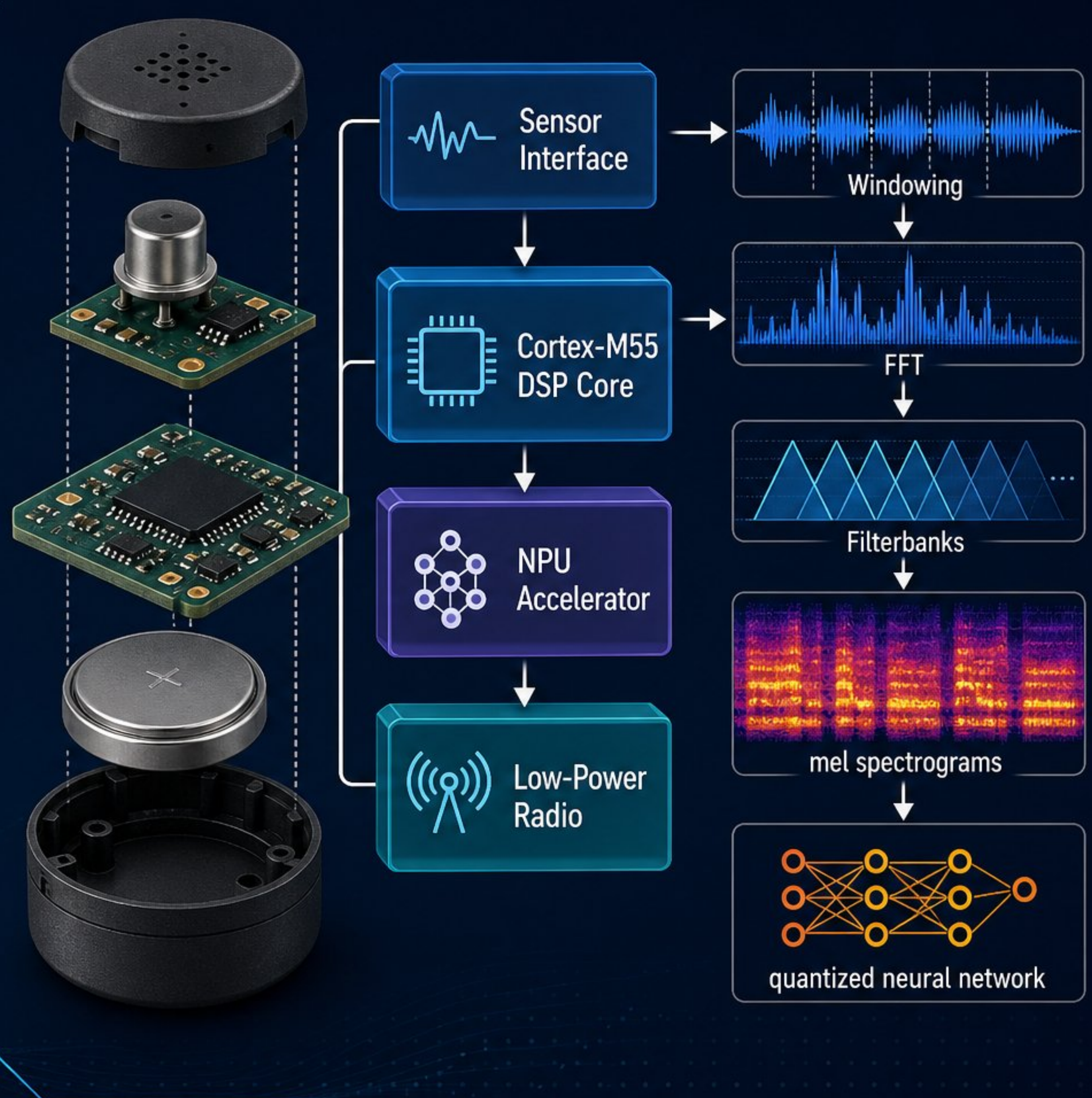


DSP in Action: Audio, Images, and Biomedicine

- **Audio:** Adaptive FIR for noise cancellation; IIR biquads for EQs and crossovers
- **Images:** 2D convolution for sharpening/blurring; frequency-domain transforms for JPEG/MPEG compression
- **Biomedicine:** Notch filters remove 50/60 Hz hum; super-resolution ultrasound breaks the diffraction limit
- **Embedded:** Real-time FFT and snore detection on ESP32 with MEMS microphones

Into the Tiny World: DSP on Microcontrollers and Edge AI

- - **Embedded DSP:** Cortex-M4/M33/M55 MCUs, dedicated cores, and NPUs for battery-powered sensors
- - **STM32U3's HSP:** 12× faster FFT vs. Cortex-M33, enabling coin-cell vibration monitors
- - **TinyML pipeline:** Windowing, FFT, filterbanks → mel spectrograms → quantized neural network
- - **Developer tools:** Edge Impulse, STM32Cube.AI, CMSIS-DSP, Python scipy.signal for prototyping



The DSP Practitioner's Checklist and Next Steps

- Know signal's max frequency, sample at > 2×, place anti-alias filter before ADC
- Choose bit depth for dynamic range; design FIR/IIR to spec, accounting for quantization
- Implement carefully: check for overflow, limit cycles, and coefficient precision
- Reconstruct cleanly with anti-imaging filter after DAC
- Next: master the FFT, explore adaptive filtering, and build a Python DSP sandbox