DIGITAL MEETS ANALOG

CHAPTER 13

Here we meet the major subject of conversion between analog and digital signals – analog-to-digital (A/D) and digital-to-analog (D/A) converters (ADCs and DACs) – as well as the important "mixed-signal" phase-locked loop (PLL). And we cannot resist a look at the fascinating topic of pseudorandom noise generation.

We live in a largely analog (continuous) world – of sounds, images, distances, times, voltages and currents, and so on – which would seem to call for analog circuits (oscillators, amplifiers, filters, combiners, etc.). But we live also in a partly digital (discrete) world – of numbers and arithmetic, of text and symbols, and the like – which would seem to call for digital circuits (arithmetic logic and storage, etc.). And that's how it was, for many years: analog amplifiers and filters for audio and video; analog oscillators, tuned circuits, and mixers for radio and television; and even *analog computers*, for solving differential equations[1] or for real-time control of flight or weaponry. Meanwhile digital techniques (initially with mechanisms and relays, then with vacuum tubes, followed by discrete transistors, small-scale ICs, and finally the large and fast microprocessors with a billion+ transistors that we take for granted) were used for computational tasks like keeping track of money, and of words.

But the almost miraculous improvements in the speeds and densities of purely digital electronics have produced a major paradigm shift, namely, the use of digital conditioning and processing for nearly every "analog" quantity. For example, audio engineers now digitize the individual microphone signals at the time of recording, and perform all subsequent mixing and conditioning (e.g., the addition of reverberation) as arithmetic on those numbers; the same goes for digital video. And at the everyday level, digital techniques have invaded our lives: the authors' bathroom scales indicate to 0.1 pound (sometimes to our regret) – that's a resolution[2] of a part per thousand; our porch light is switched on and off by a digital wall switch that follows the seasonal variation of dusk and dawn; and our automobiles depend on a digital bus, to which are connected some 50 or more embedded digital controllers for functions like engine control and diagnostics, braking, air bags, entertainment, climate control, and so on.

The bottom line is that A/D and D/A conversion techniques have become central to every aspect of analog measurement and control. This is important stuff, and it is the major subject of this chapter. Let's go at it.

Our treatment of the various conversion techniques is not aimed at developing skill in converter design itself. Rather, we try to point out the advantages and disadvantages of each method, because in most cases the sensible thing is to buy commercially available chips or modules, rather than to build the converter from scratch. An understanding of conversion techniques and idiosyncrasies will guide you in choosing from among the thousands of available units.

13.1 Some preliminaries

13.1.1 The basic performance parameters

Before getting into lots of detail, we'd like to summarize the important performance parameters that you need to keep in mind when choosing ADCs and DACs. Knowing what you need makes it a lot easier to find what you want.

Digital-to-analog converters

Resolution: number of bits

Accuracy: monotonicity; linearity; dc stability

Reference: internal or external; multiplying DAC (MDAC)?

Output type: voltage output or current output

Output scaling: unipolarity or bipolarity; V_{out} ranges; I_{out} compliance

Speed: settling time; update rate

[1] There's a nice example of this in the section on Analog Function Circuits in Chapter *4x*: modeling the fascinating chaotic behavior of the system of nonlinear differential equations devised by Lorenz.

[2] To be distinguished from *accuracy* – recall the discussion in §5.1.1. Although the bathroom scale reads with a *resolution* of 0.1 pound, its accuracy is likely poorer (perhaps to ±1 pound), with some drift over time and temperature.

Quantity: single or multiple DACs/pkg
Digital input format: serial (I^2C, SPI, or a variant) or parallel
Package: module, through-hole, or various surface-mount packages
Other: glitch energy; power-on state; programmable internal digital scaling

Analog-to-digital converters
Resolution: number of bits
Accuracy: monotonicity; linearity; dc stability
Reference: internal or external
Input scaling: unipolarity or bipolarity; voltage range
Speed: conversion time and latency
Quantity: single or multiple ADCs/pkg
Digital-output format: serial (I^2C, SPI, or a variant) or parallel
Package: module, through-hole, or various surface-mount packages
Other: internal programmable gain amplifier (PGA); spur-free dynamic range (SFDR)

13.1.2 Codes

At this point you should review §10.1.3 on the various number codes used to represent signed numbers. Offset binary and 2s complement are commonly used in A/D conversion schemes, with sign-magnitude and Gray codes also popping up from time to time. Here is a reminder:

	Offset binary	*2s Complement*
+Full scale	11111111	01111111
+Full scale−1	11111110	01111110
↓	↓	↓
0+1 LSB	10000001	00000001
0	10000000	00000000
0−1 LSB	01111111	11111111
↓	↓	↓
−Full scale+1	00000001	10000001
−Full scale	00000000	10000000

13.1.3 Converter errors

The subject of ADC and DAC errors is a complicated one, about which whole volumes could be written. According to Bernie Gordon at Analogic, if you think a high-accuracy converter system lives up to its claimed specifications, you probably haven't looked closely enough. We won't go into the application scenarios necessary to support Bernie's claim, but it's worth a first look at the four most common types of converter errors: offset error, scale error, nonlinearity, and nonmonotonicity, nicely illustrated in the self-explanatory Figure 13.1. Rather than boring you with a long-winded discussion, though, we'll move directly to a description of D/A converter techniques and capabilities. Then we'll revisit the business of converter errors (§13.4), which will make a lot more sense in context.

13.1.4 Stand-alone versus integrated

Sometimes an ADC or DAC (or both) is integrated into a fancier IC. The most common example is the microcontroller (Chapter 15), where you frequently see both ADCs and DACs integrated on the same chip as the processor and its other I/O peripherals. As far as we can tell, the least-expensive stand-alone ADC costs significantly more than the least-expensive microcontroller-*with*-ADC.[3] Microcontrollers are fond of integrating lots of useful peripherals, along with program and data memory, so that you've got essentially a "system-on-a-chip." Be aware, though, that these converters that come bundled with inexpensive general-purpose microcontrollers do not attain the excellent performance of a good stand-alone converter: you can get 8-bit or even 10-bit performance; but you won't get 16 bits, and nothing approaching the 24-bit performance of a high-quality audio ADC.[4]

For some classes of IC, though, an integrated converter delivers excellent performance. One example is a direct digital synthesis (DDS) chip (§7.1.8), where on-chip phase counters and a sine lookup table create digital values of the synthesized sinewave output; these things can clock at speeds of 1 GHz or more, with an on-chip 14-bit (for example) DAC generating the analog output signal. Another example comes from the video world, where it's common to see digital video processing and conversion functions combined on a single high-performance IC. And in the audio business you see parts like the Cirrus CS470xx-series (their "All-In-One Audio IC" system-on-a-chip), which includes multiple 24-bit ADCs and DACs with 105 dB dynamic range, integrated onto a chip that has a 32-bit DSP (with 32 kB RAM), audio codecs and sample rate

[3] To wit: National's ADC0831 8-bit ADC costs $1.85, whereas Microchip's PIC10F with its 8-bit ADC and 2-input multiplexer costs $0.48 (both in quantity 25).

[4] A shining exception is provided by Analog Devices' series of "Analog Microcontrollers," with honest performance to 16 or 24 bits. You might think of these as consisting of a high-quality converter core, with a quiet microcontroller tacked on.

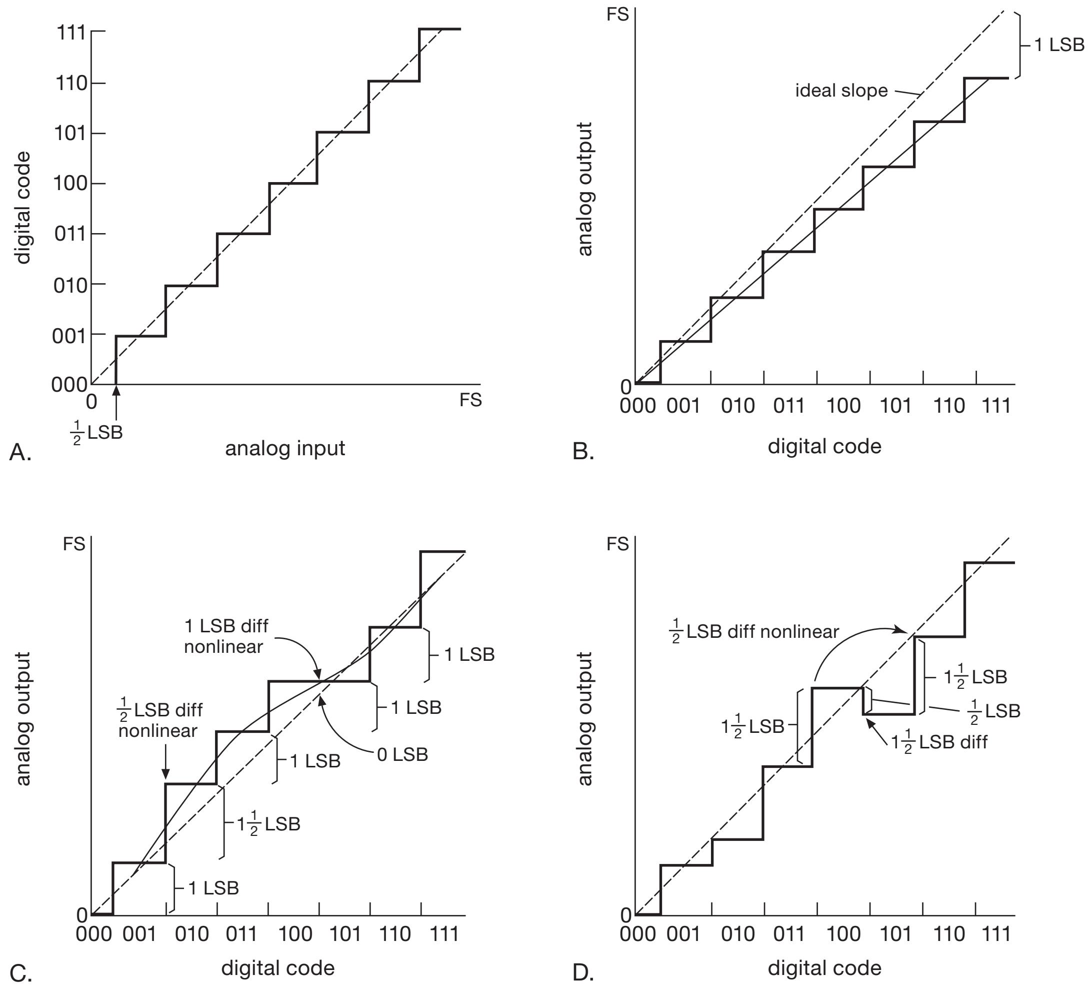

Figure 13.1. Graphs illustrating the definitions of four common digital conversion errors, for a 3-bit converter over its 8 levels from 0 to full scale (FS). A. ADC transfer curve, $\frac{1}{2}$LSB offset at zero. B. Linear, 1 LSB scale error. C. $\pm\frac{1}{2}$ LSB nonlinearity (implies 1 LSB possible error); 1 LSB differential nonlinearity (implies monotonicity). D. Nonmonotonic (must be $> \pm\frac{1}{2}$ LSB nonlinear). Used with permission of Texas Instruments Inc.

converters, digital audio ports (SPDIF), and an SPI/I^{2}C control port.

Stand-alone converters are dominant, though, in high-accuracy and high-linearity applications (voltmeters; quality audio gear). They also provide a tremendous selection range, in terms of the many parameters just listed, as compared with the rather limited selection of on-chip converters you find in microcontrollers.

13.2 Digital-to-analog converters

The goal is to convert a quantity specified as a binary (or multidigit BCD, see §10.1.3B) number to a voltage, or to a current, proportional to the value of the digital input. There are several popular methods: (a) resistor string with MOS switches; (b) *R*–2*R* ladder; (c) binary-scaled current sources; and (d) delta–sigma (and other pulse-averaging) converters. Let's take them in turns.

13.2.1 Resistor-string DACs

This method is about as straightforward as you can get. A string of 2^n equal-value resistors is connected between a stable voltage reference and ground, creating a very tall voltage divider; and a set of MOSFET analog switches is used to route the selected tap's voltage to an output voltage buffer (Figure 13.2). The figure shows the configuration of TI's impressive DAC8564, a quad 16-bit DAC (four