Behzad Razavi

The Current-Steering DAC

Digital-to-analog converters (DACs) find application in many systems, including communication transmitters and consumer electronics. Among various DAC realizations, the current-steering topology offers the highest speed and becomes the de facto solution at gigahertz frequencies, especially if the analog output must be delivered to a resistive load. In this article, we study this DAC's design principles.

Basic Topology

We wish to convert an N-b digital signal, D_{in}, to an analog current, I_{out}. This can be accomplished as illustrated in Figure 1(a), where each input bit controls a current that is binarily weighted with respect to a unit value, I_u. Here, D_1 denotes the least significant bit (LSB) and D_N the most significant bit (MSB). The current sources are scaled up by a factor of two from one bit to the next, yielding

$$I_{out}=D_N(2^{N-1}I_u)+\cdots+D_2(2I_u)+D_1I_u. \quad (1)$$

This circuit is an example of a simple binary-weighted DAC. We can also call it a current-switching—but not a current-steering—implementation.

An important advantage of this DAC over other types is its ability to drive resistive loads with no need for a buffer. This property proves crucial if the DAC must drive a transmission line, as in wireline systems, or if the load contains a resistive component, as in displays and optical modulators.

The current-switching structure shown in Figure 1(a) suffers from dynamic errors. As depicted in Figure 1(b), when a switch turns off, the top terminal voltage of its corresponding current source collapses to zero. Thus, the next time that this branch is enabled, the (nonlinear) capacitance at this terminal must charge up, drawing a significant transient current from the output node. Moreover, since switching actions change the total current carried by the array, the ground voltage experiences large fluctuations in the presence of parasitic series inductances, such as those due to bond wires.

Both of these effects can be greatly suppressed through the use of current steering (Figure 2). Here, the tail current is steered to the left or the right by each differential pair, causing only a small voltage excursion at node X. Also, since the total array current is relatively constant, the ground bounce is much smaller. Of course, another advantage of this configuration is that it naturally provides differential outputs.

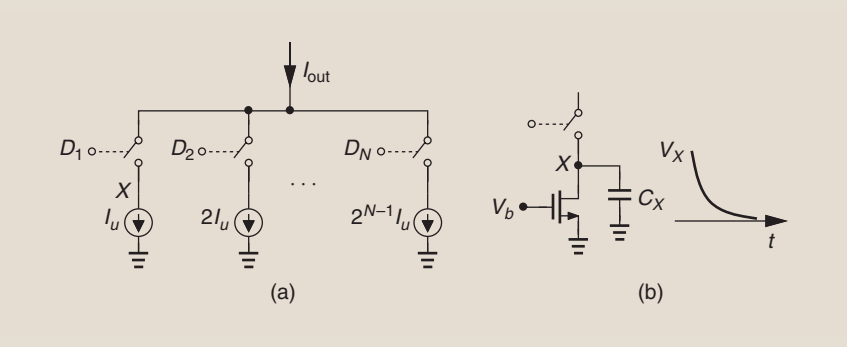

FIGURE 1: (a) A simple binary-weighted current-switching DAC and (b) the problem of discharge at X when the switch is off.

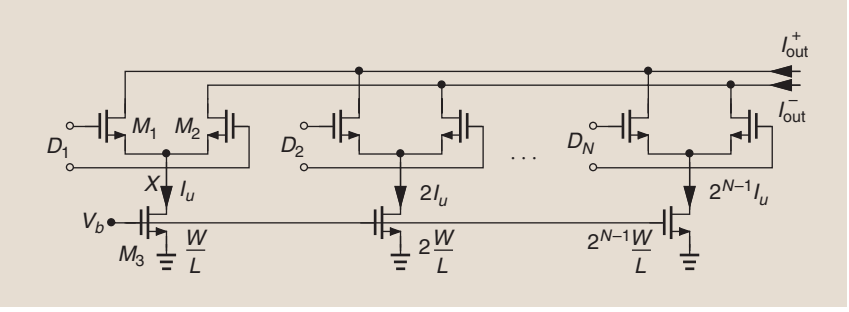

FIGURE 2: A binary-weighted current-steering DAC.

Digital Object Identifier 10.1109/MSSC.2017.2771102
Date of publication: 31 January 2018

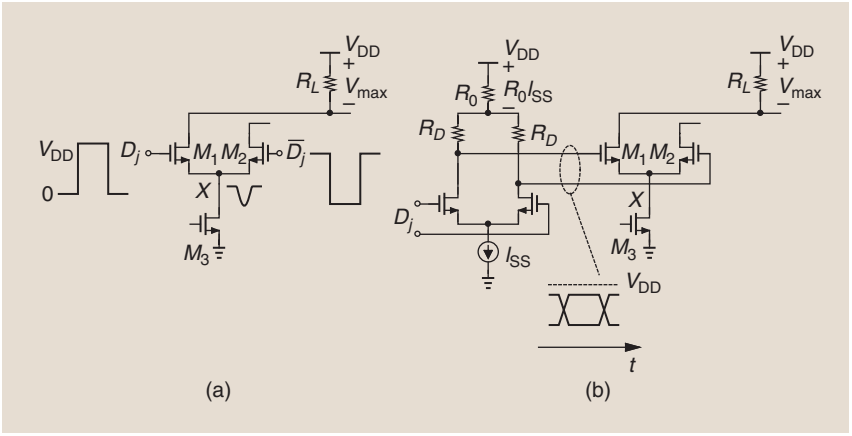

FIGURE 3: A DAC cell with (a) rail-to-rail or (b) moderate digital swings.

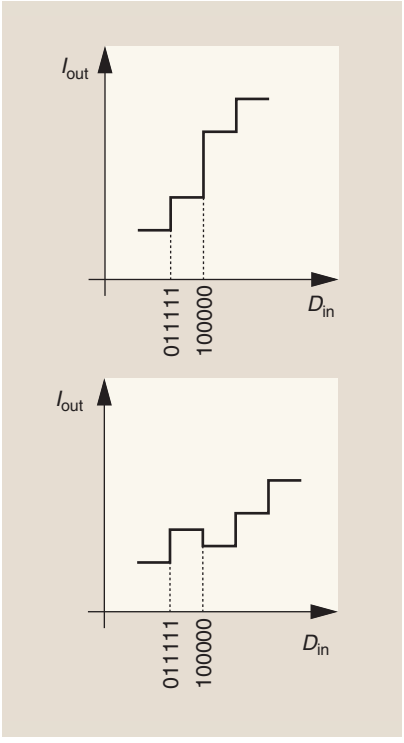

FIGURE 4: Large jumps or nonmonotonicity in binary-weighted DAC characteristics due to mismatches.

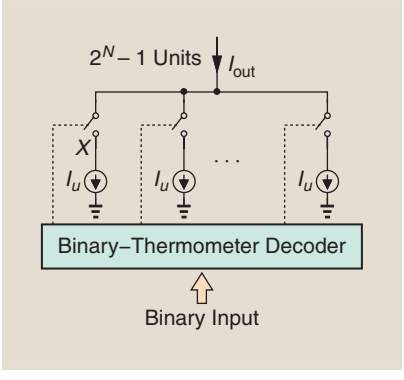

FIGURE 5: A segmented DAC.

For proper matching among the current sources, we design a unit cell comprising a current source and a differential pair and repeat this cell to form larger cells. That is, cell number j consists of 2^{j-1} unit cells in parallel, and the entire DAC contains $2^N - 1$ unit cells.

One drawback of current-steering DACs is their limited output voltage compliance. In Figure 2, for example, the differential-pair transistors must operate in saturation (as explained below), and, therefore, at least two drain-source voltages are subtracted from the supply, V_{DD}.

Another difficulty in the design is the choice of the digital input voltage excursions. The most convenient are rail-to-rail swings, but, as shown in Figure 3(a), such a choice 1) limits the analog output voltage range V_{max} to one transistor threshold if M_1 must remain in saturation and 2) leads to large dips in V_X during the transitions of D_j and $\overline{D_j}$. In other words, we would prefer only a moderate swing for the digital inputs, with a maximum level less than V_{DD} so as to allow a greater V_{max}. Such swings call for another differential pair [Figure 3(b)] and hence substantial power consumption.

The Need for Segmentation

The binary-weighted arrays in Figure 1(a) or Figure 2 can face undesirable jumps in their output when the digital input goes from 011⋯1 to 10⋯0. We observe that the output current is provided by the $I_u, 2I_u, \ldots,$ and $2^{N-2}I_u$ current sources for the former code and by the $2^{N-1}I_u$ current source for the latter. The difference is nominally equal to 1 LSB $= I_u$, but, with mismatches present in the circuit, it is possible that the sum of the former group is substantially different from $2^{N-1}I_u - I_u$. As a result, the DAC input–output characteristic can exhibit a large error or nonmonotonicity at this transition (Figure 4). The fundamental difficulty here is that, at this "major carry" transition, a group of current sources turns off and a new current source turns on.

The foregoing issue can be avoided by "segmentation." For an N-b DAC, we still incorporate $2^N - 1$ unit cells but apply a different switching sequence. As shown in Figure 5, for a binary input 000⋯01, one cell is activated; for 00⋯10, two; for 00⋯11, three; etc. We say the cells are driven by a "thermometer code." For example, as the binary input goes from 01 to 10 to 11, the corresponding thermometer code changes from 0001 to 0011 to 0111.

The segmented architecture avoids the jumps shown in Figure 4 because, at the major carry transition, it simply turns on one more LSB cell, rather than turn off one group of current sources. Thus, the output changes monotonically, and the jump is not much different from 1 LSB as the new cell has *some* matching with respect to the previous cells. In practice, of course, each cell is based on a current-steering structure.

Partial Segmentation

We have seen that the number of unit cells is the same for binary and segmented architectures and becomes prohibitively large at high resolutions. For example, a 10-b DAC would require 1,023 cells, facing severe area and routing issues. We note, however, that the matching requirements are more relaxed for the LSB current sources: even if the first and second LSB currents have a mismatch of 10%, the overall characteristic can still reach 10-b precision. Let us consider the following approach: rather

than copy currents by means of unit cells, we seek a method of *dividing* currents by binary factors. For example, we can keep doubling the length of the current-source transistors, as shown in Figure 6(a). However, the effective length does not double, creating significant errors. Instead, we place identical transistors in series [Figure 6(b)]. The resulting architecture is called a "partially segmented DAC" to emphasize that only the MSB section is segmented.

The exact partitioning of the DAC into segmented and binary sections depends on the matching properties of the transistors; the binary array can still suffer from effects shown in Figure 4. In a typical design, we use binary weighting for the first three or four LSBs and segmentation for the remaining bits.

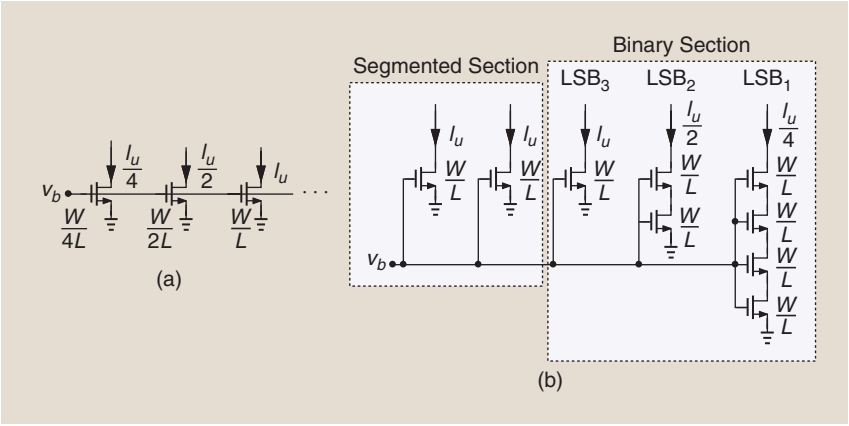

FIGURE 6: (a) Binary weighting by doubling transistor lengths and (b) a partially segmented DAC employing transistors in series.

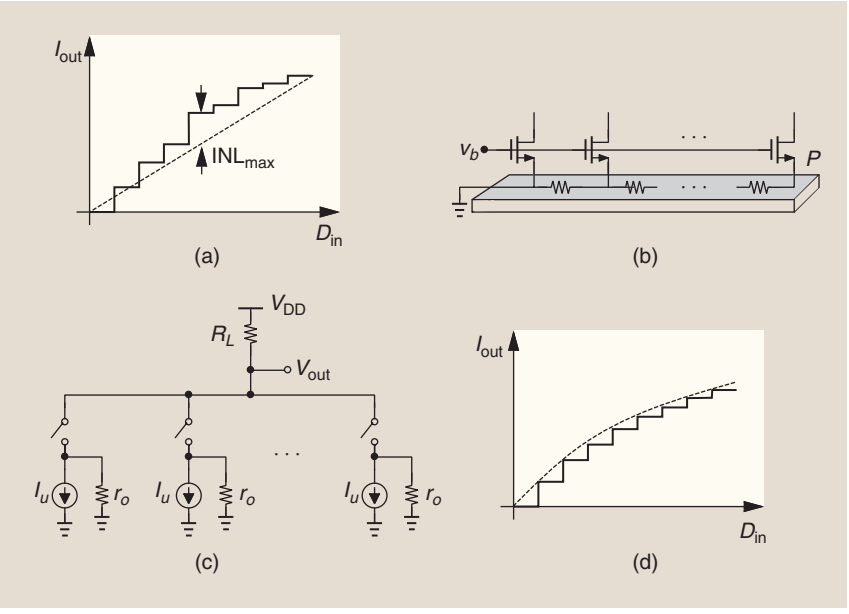

FIGURE 7: (a) An illustration of INL, (b) the effect of ground line IR drops, (c) the effect of output impedance of current sources, and (d) a compressed characteristic arising from (c).

Static Errors

Current-steering DACs must deal with three types of static errors. First, the random mismatches among the current sources distort the input–output characteristic. These mismatches accumulate and primarily manifest themselves in the form of integral nonlinearity (INL) in segmented topologies. Illustrated in Figure 7(a), the INL is defined as the error between the actual characteristic and a straight line passed through its points. Second, the voltage drop along the ground line traveling to the current cells can cause significant deterministic nonlinearity. As depicted in Figure 7(b), if a large number of cells inject current into a long ground line, the voltage at the farthest point from ground, P, can reach tens of millivolts. With a nominal overdrive voltage of, say, 200 mV for the current sources, the ground drop introduces excessive nonlinearity.

The third static error relates to the finite output resistance of the cells if the DAC must drive a resistance. From Figure 7(c), we observe that the incremental resistance at the output node varies from $R_L || r_O$ when only one cell turns on to $R_L || (r_O/M)$ when all M cells turn on. In other words, the first current source produces a voltage change equal to $I_u(R_L || r_O)$ and the last, $I_u[R_L || (r_O/M)]$. Due to this code-dependent output resistance, the input–output characteristic exhibits compression as D_{in} increases [Figure 7(d)]. It can be shown that the maximum INL arising from this effect is given by $I_u R_L^2 M^2/(4r_O)$, which, normalized to the full-scale output voltage, is approximately equal to $MR_L/(4r_O)$. For example, if $M = 1024$ and $R_L = 50\,\Omega$, then r_O must exceed 12.8 MΩ for the INL to remain below 0.1%. Such an extremely high output impedance is difficult to obtain in practice.

Since differential operation suppresses even harmonics, we expect the differential counterpart of the array in Figure 7(c) to achieve a higher linearity or, for a given INL, require a less stringent output resistance.

Dynamic Errors

The current-steering DAC of Figure 2 also suffers from dynamic errors, and hence greater distortion, at high output frequencies. We examine three such errors here.

The tail node capacitance in Figure 2 introduces nonlinearity—even though the voltage swing at this node

is typically lower than 100 mV. The capacitance, C_X, degrades the performance through two mechanisms [1]. First, it simply appears in parallel with the tail current source, lowering the unit cell output impedance and exacerbating the code-dependent output impedance described in the previous section. It can be shown that the nth harmonic at the output has a relative magnitude given by $[MR_L/(4\,|\,Z_u\,|)]^{n-1}$, where Z_u denotes the complex output impedance of the unit cell [1].

The second mechanism related to the tail capacitance involves charge transfer from one clock cycle to another [1]. As illustrated in Figure 8, when $D_j = 1$, C_X charges to a voltage that, due to r_O, depends on $V^+_{out}(t_1)$. Now, some clock cycles later, D_j goes to zero and V_X must now change so as to track $V^-_{out}(t_2)$, which can be very different from $V^+_{out}(t_1)$. For this change to occur, C_X must receive some charge equal to $C_X(V_2 - V_1)$ from V^-_{out}, causing a dynamic error at the output.

Another type of dynamic error arises in Figure 2 from the coupling of the tail nodes to the bias line, V_b, through the gate-drain capacitance of the current sources. Depicted in Figure 9, the resulting jump in V_b disturbs the bias current of the cells, thus increasing the output settling time. The principal difficulty here is that even a few millivolts of change in V_b translate to an output disturbance of many LSBs. One can contemplate the use of a bypass capacitor at V_b, but this capacitor also slows down the settling at this node. Another remedy is to add a cascode device atop the current sources while losing voltage headroom.

The third dynamic error in current-steering DACs stems from mismatches between the times at which the data edges reach the unit cells. To minimize this error, each cell is preceded by a latch, but clock skews and random mismatches between the latches still limit the performance [2].

Design Procedure

The design of a current-steering DAC begins with the unit cell. We must size and bias the tail current source so as to guarantee a maximum static INL of lower than 1 LSB due to both random mismatches and the code-dependent output resistance. It can be shown that, for a random change of σ_I in the unit current, I_u,

$$\mathrm{INL}_{\max} = \frac{\sigma_I}{2I_u}\sqrt{2^N} \ \ \mathrm{LSB}, \qquad (2)$$

where N is the resolution [3]. We also know that for two nominally identical MOS current sources that have relatively large dimensions, $\sigma_I/I_u = 2\sigma_{VTH}/(V_{GS} - V_{TH})$, where σ_{VTH} denotes the threshold mismatch, and $\sigma_{VTH} = A_{VTH}/\sqrt{WL}$, where A_{VTH} is a process-dependent parameter, e.g., around 6 mV·μm. We must, therefore, choose WL and $V_{GS} - V_{TH}$ large enough to ensure the random $\mathrm{INL}_{\max}$ is lower than 0.5 LSB. In a typical design, $V_{GS} - V_{TH}$ is limited to 200–300 mV, imposing a large unit transistor if resolutions of 8 b or higher are sought. For example, with $N = 10$, $V_{GS} - V_{TH} = 200$ mV, and $A_{VTH} = 6$ mV·μm, we have $WL = 1.8\ \mu\mathrm{m}^2$. We then choose a large value for L to minimize the nonlinearity due to the output resistances. If the cascode comprising M_1 and M_3 in Figure 2 still does not yield $\mathrm{INL}_{\max} = 2^N R_L/(4r_O) < 0.5$ LSB, we can introduce a cascode device atop M_3.

To ensure fast switching and minimal capacitance at the tail node, the switching transistors are designed with the minimum channel length. We note that C_X Figure 2 is given by $C_{GS1,2} + C_{GD3} + C_{DB3}$. The width of M_1 and M_2 is chosen so as to obtain a small overdrive voltage, around 50–100 mV. As mentioned above, this pair is preceded by a latch in every cell.

The Matrix Architecture

Suppose we wish to design an 8-b fully segmented DAC. How do we arrange the 255 cells while distributing the data, clock, and power with minimal parasitics? Proposed by [4], an elegant approach arranges the unit cells in a compact matrix, making the distribution much more manageable than in a linear array. As illustrated in Figure 10, the architecture consists of a column decoder, a row decoder, and 2^N unit cells, each containing local logic, a current source, and a switching pair. (In high-speed designs, a latch also appears between the logic and the pair.) We describe the operation for a resolution of 6 b as an example. The input binary data $D_6D_5D_4D_3D_2D_1$ is decomposed into two binary words

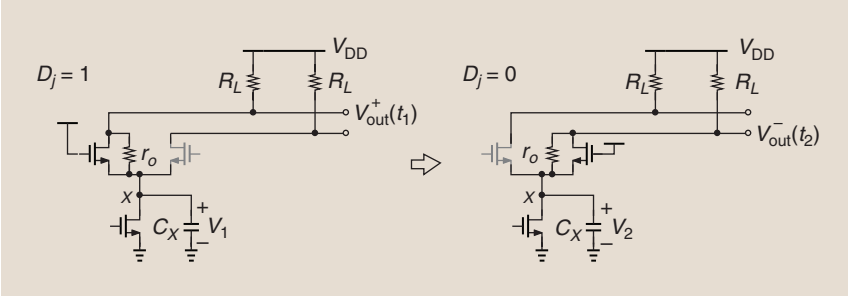

FIGURE 8: Dynamic distortion due to the dependence of V_X on output voltages.

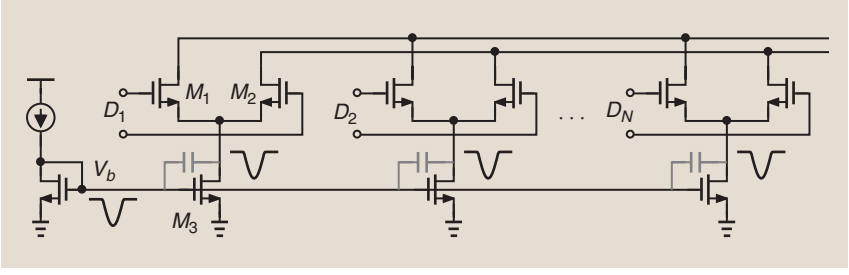

FIGURE 9: A long settling time due to feedthrough of jumps at the tail node to the bias line.

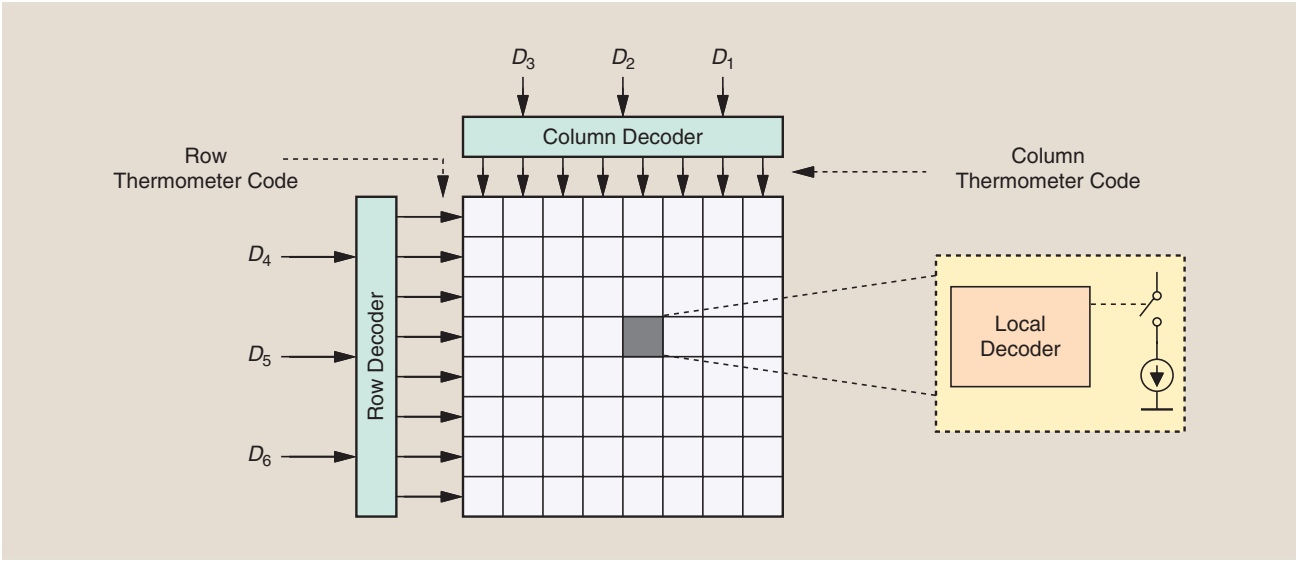

FIGURE 10: The matrix DAC architecture.

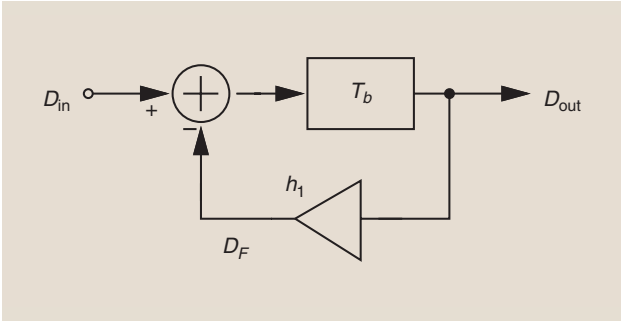

FIGURE 11: A simple DFE loop.

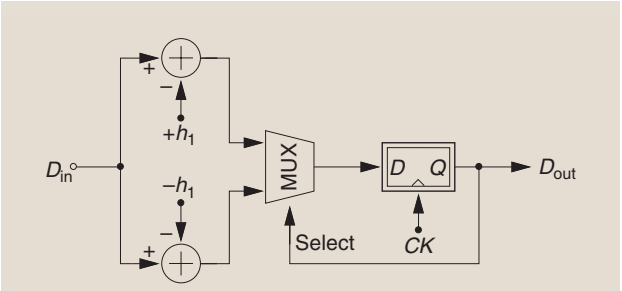

FIGURE 12: An unrolled DFE architecture.

$D_6D_5D_4$ and $D_3D_2D_1$, with one applied to the row decoder and the other to the column decoder. These words are converted to thermometer codes that travel horizontally and vertically within the matrix. Each cell senses the row and column thermometer code values to determine whether the current should be on or off. The local decoder senses the thermometer codes of two consecutive rows and one column to distinguish among three cases: 1) all cells in a row are on, 2) all cells in a row are off, and 3) some cells in a row are on [4].

Questions for the Reader

1) By what factor is the INL of a differential current-steering DAC lower than that of a single-ended topology if only the finite output impedance of the current sources is considered?
2) In the matrix architecture of Figure 10, each row experiences the same gradient from left to right. If each cell current is higher than the one to its left by ΔI, what is the maximum INL?

Answers to Last Issue's Questions

1) Can the delay stage and the slicer in Figure 11 be realized as a single limiting differential pair?

   If the total delay is chosen equal to 1 UI, yes, it can. However, the gain may not suffice to amplify the summer output to logical levels. Also, the delay of such an asynchronous stage varies with process and temperature.
2) Can the unrolled DFE of Figure 12 accommodate a second tap?

   Yes, it can. We must add another flip-flop after the first, scale its output according to the value of the second tap, and inject the result to both summers. Alternatively, we can return the result to a summer inserted at the very input.

References

[1] P. Palmers and M. S. J. Steyaert, "A 10-bit 1.6-GS/s 27-mW current-steering D/A converter with 550-MHz 54-dB SFDR bandwidth in 130-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 57, pp. 2870–2879, Nov. 2000.

[2] K. L. Chan and I. Galton, "A 14b 100MS/s DAC with fully segmented dynamic element matching," in *Proc. Int. Solid State Circuits Conf. Dig. Tech. Papers*, Feb. 2006, pp. 214–215.

[3] C.-H. Lin and K. Bult, "A 10-b 500MS/s CMOS DAC in 0.6 mm^2," *IEEE J. Solid-State Circuits*, vol. 33, pp. 1948–1958, Dec. 1998.

[4] T. Miki, "An 80-MHz 8-bit CMOS D/A converter," *IEEE J. Solid-State Circuits*, vol. 21, pp. 983–988, Dec. 1986.