Optimization techniques for decoding logic design in digital-to-analog converters

Kyaw Kyaw
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd

Recommended Citation
https://lib.dr.iastate.edu/rtd/19148

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Optimization techniques for decoding logic design in digital-to-analog converters

by

Kyaw Kyaw

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:
Randall L. Geiger, Major Professor
Degang Chen
Stuart J. Birrell

Iowa State University
Ames, Iowa
2005
This is to certify that the master’s thesis of

Kyaw Kyaw

has met the thesis requirements of Iowa State University
# TABLE OF CONTENTS

**ABSTRACT**

**CHAPTER 1. INTRODUCTION**
- Motivation 1
- Thesis Organization 2

**CHAPTER 2. DIGITAL-TO-ANALOG CONVERTER ARCHITECTURES**
- Overview 3
- Voltage Division Architectures 3
  - Resistor Ladder DACs 3
  - Folded Resistor-string DACs 5
  - Multiple Resistor-string DACs 6
- Current Steering Architectures 8
- Charge Redistribution Architectures 9
- Hybrid Architectures 10

**CHAPTER 3. DECODING SCHEMES AND THEIR RELATIVE PERFORMANCE**
- Overview 12
- Binary-weighted Architectures 12
- Thermometer-coded Architectures 13
- Segmented Architectures 13
- DNL Performance of Decoding Schemes 14
- Yields for Decoding Schemes 17
- Digital and Analog Area Requirements for Different Decoding Schemes 20
- Conclusion 21

**CHAPTER 4. OPTIMIZATION TECHNIQUES FOR DECODING LOGIC**
- Overview 23
- Row/Column Decoding Logic 23
- Nested Row/Column Decoding Logic 25
- Linear-step Decoding Scheme 29
- Multi-segmented Linear-step Decoding Scheme 34
- Conclusions 36

**CHAPTER 5. A 10-BIT DIGITAL-TO-ANALOG CONVERTER USING MULTI-SEGMENTED LINEAR-STEP DECODER**
- Overview 38
- Architecture Selection 38
- Analog Design Aspects 38
  - Power Consumption 38
  - Matching Requirement 40
  - Gradient Errors 41
Output Impedance 43
Digital Design Aspects 44
Simulation and Testing Results 44
  Prototyping and Simulation 44
  Experimental Results 46
  Static Performance 47
  Dynamic Performance 49
Conclusions 50

REFERENCES 52

ACKNOWLEDGEMENTS 54
ABSTRACT

High resolution and high speed digital-to-analog Converters are important building blocks in many telecommunications and digital signal processing circuits. To achieve high accuracy, thermometer-coded structures are almost always used either as a whole or part of a segmented architecture. However, the design of a binary-to-thermometer decoder at high resolutions to achieve high decoding speed is non-trivial and is often the bottleneck in the system. The advantages of having more thermometer-coded bits in the architecture are high linearity performance, relaxed matching requirements and higher yield. However, large binary-to-thermometer decoder usually takes a lot of area and suffers in decoding speed. Some practical ways to reduce both the time delay and the digital logic involve with the large decoders are discussed. A 10-bit digital-to-analog converter was designed and fabricated using multi-segmented linear-step decoding logic and experimental results are presented.
CHAPTER 1

INTRODUCTION

Motivation

Digital-to-analog converters are crucial components in many communication and signal processing systems. The data converter design often limits the performance of these systems. As semiconductor fabrication process technology continues to advance according to Moore’s Law, the shrinking feature sizes of CMOS devices offers potential for a reduction in the size of integrated circuits along with improvements in performance. However, in data converter design, where matching characteristics of components is a major concern, designers have realized few benefits, if any, from the continuing shrinking feature sizes.

Figure 1.1 – Moore’s Law on feature sizes (source: Intel)
As will be discussed later, the decoding method used in data converters usually plays a key role in the performance of such systems. Thermometer decoding, which provides favorable differential nonlinearity performance, is almost always employed for the MSB blocks in high resolution, high speed systems. Being digital in nature, the decoder part of a data converter can exploit the benefits of shrinking feature sizes. A better design of the decoder, even at the expense of a larger number of gates, could allow the designer to use more thermometer-coded bits in the DAC and thus relax the matching requirements in the analog part of the circuit. Performance improvements could also be obtained by designing a thermometer-coded decoder in a way so will allow a reduction in the total gate count.

**Thesis Organization**

An overview of popular digital-to-analog architectures and their relative advantages and disadvantages is presented in Chapter 2. In Chapter 3, the relative performances of three different decoding schemes are discussed. Methods of optimizing the thermometer decoder are discussed in Chapter 4. Included in this discussion is a new multi-segmented linear-step decoding scheme. To demonstrate the functionality of the multi-segmented linear-step decoding scheme, a 10-bit prototype DAC using multi-segmented linear-step decoding logic was fabricated and tested. Design details and experimental results for this prototype DAC are discussed in Chapter 5.
CHAPTER 2

DIGITAL-TO-ANALOG CONVERTER ARCHITECTURES

Overview

There are several different topologies that are used by designers to realize digital-to-analog converters (DAC). DACs can generally be classified into four main classes; voltage division, current steering, charge redistribution and hybrid topologies. Each class offers both advantages and disadvantages. Invariably tradeoffs from one class to another are made in terms of monotonicity, digital decoder circuit, and speed requirements. This chapter briefly discusses the various topologies and describes the pros and cons of each.

Voltage Division Architectures

Resistor Ladder DACs

Resistor ladder DACs employ a string of equally sized resistors, [1]. Figure 2.1 shows a typical resistor ladder network configured into a resistor ladder DAC. The output buffer is optional. Typically an, n-bit architecture requires $2^n$ resistors connected in series between $+V_{ref}$ and $-V_{ref}$. A tap point is selected based on the digital input code, $B_{in}$. A tree-like switch network is often used as an analog MUX for selecting the desired tap voltage. The switch network is normally implemented by using a single n-channel CMOS transistor or by using transmission gates individually made up of a p-channel device and an n-channel device.
One major advantage of the resistor ladder architecture is its simplicity. The tree-like decoding circuitry is quite simple. The analog output levels at the input to the buffer are inherently monotonic and this monotonicity is generally maintained at the output of the buffer even when the offset voltage of the buffer is considered [2]. However, there are some drawbacks in the design shown in Figure 2.1. First, the delay in the switch network limits the conversion speed. This issue can be partially resolved by using a binary to thermometer decoder as shown in Figure 2.2. By using such a simple decoder, the delay in the switch network can be greatly reduced but additional area may be required for the decoder circuit and signal routing, particularly if the resolution level is high.

Figure 2.1 – Resistor Ladder DAC
Another disadvantage of the resistor ladder DAC is the component count. For ‘n’ bit resolution, it requires $2^n$ resistors which can be relatively large at higher resolution levels.

![Resistor Ladder DAC with Decoder Circuit](image)

**Legend:**
- $V_{ref}$
- $R$
- $V_{out}$
- $b_1$
- $b_2$
- $b_3$

**Figure 2.3 – Resistor Ladder DAC with Decoder Circuit**

**Folded Resistor-string DACs**

In order to address the requirement for a large decoder circuit, the authors in [3] proposed a folded resistor string topology as shown in Figure 2.3. In this approach, a decoding network which is very similar to that employed in random access memories is used. The most significant bits of the digital input code are used to select a single row in the structure and the least significant bits are used to select a single column.
The major advantages of this architecture are the reduced capacitive loading and the amount of digital decoding required. In the structure shown in Figure 2.2, a total of $2^n$ transistors junctions are connected together resulting in a large capacitive load. On the contrary, only $2 \cdot \sqrt{2^n}$ transistors are connected to the output bus, making this topology more suitable for higher speed applications.

![Diagram of a 4-bit folded resistor-string converter](image)

**Figure 2.3 – A 4-bit folded resistor-string converter**

**Multiple Resistor-string DACs**

A multiple resistor-string architecture, such as that used in [4], reduces the number of components required. In this topology, two resistor strings are used. A buffer is often used
with this approach to keep the second resistor string from loading the first. The second strong provides an interpolation between two adjacent nodes in the first resistor string as shown in Figure 2.4.

![Multiple Resistor-string DAC Diagram](image)

**Figure 2.4 – Multiple Resistor-string DAC**

In this type of data converter, the MSB bits determine which two adjacent nodes of the first resistor string should be connected to the buffers inputs. The LSB bits are then utilized to interpolate between the resistors in the second string where the output is determined. In the structure shown, some extra logic is required in the interpolator stage as the high and low voltages switch between top and bottom buffers. This approach needs only
$2 \times 2^{n/2}$ resistors and hence greatly reduces the number of component required. This topology is attractive for high resolution architectures. The matching requirement on the resistors in the second string is greatly reduced as well since it is used to decode the LSB bits. The limitations of this approach lies with the buffers. For high speed applications, they must be fast, they must have low noise, and they must have matched offset voltages which are not dependent on the input voltages. The design of the buffers becomes non-trivial when using this architecture to design high resolution DACs that operate at high speed architectures.

**Current Steering Architectures**

Just like the voltage division architecture, the current steering DAC operates by steering or dividing a reference current and selecting among certain current outputs. As shown in Figure 2.5, a reference current source is replicated in each current cell of the DAC, and each cell is switched on or off based on the input code. The weight $i_k$, where $\sum_{k=1}^{L} \sum_{k=1}^{L} 2^{n-1}$, on each cell can be varied according to the decoded signal $d$ used. The area requirement of the cells for matching and monotonicity depends on the weight difference of the current cells. For example, the thermometer coded decoding with for all current cells will guarantee monotonicity for some switching sequences. The decoding schemes and the advantages associated with current steering architectures are discussed in Chapter 3.

The major advantage of current steering architectures is the speed, [5]. As no output buffer which has limited bandwidth in the operational amplifier is required, the speed of a current steering DAC is dominated by the speed of the decoder circuits and the settling time.
of the output switches. The main limitation with the current steering architectures is the glitches which occur when switching between different signals, [2]. The magnitude of the glitch depends on the weight difference between the current cells.

![Current Steering DAC Architecture](image)

**Figure 2.5 – Current Steering DAC Architecture**

**Charge Redistribution Architecture**

Another popular type of DAC is the charge distribution architecture [6] shown in Figure 2.6. The bottom plates of the capacitors are all connected either to the ground or to $V_{ref}$, injecting charge of $Q = i_k \cdot C_{nom} \cdot V_{ref}$ to the output node. As discussed previously in current division architecture, the size of individual capacitors depends on the decoder logic being employed.

The circuit shown in Figure 2.6 operates in two stages. During clock phase $\phi_1$, the selected capacitors are connected to $V_{ref}$ and being charged up. In the mean time, in the output circuit, the feedback capacitor is being discharged and compensation capacitor $C_{com}$ is
being charged to negative of input offset of the amplifier. In phase $\phi_2$, the cumulative charges on the capacitor network are transferred on to the feedback capacitor. Also, the input offset stored on the compensation capacitor is subtracted from the output voltage. Clearly, the timing and control logic for charge sharing digital-to-analog converters is significantly more complex than that of resistor-ladder topologies. They also suffer from the same mismatch problems (both linear gradient and random mismatches occur) as resistor-string DACs, and decent size capacitors in CMOS takes a considerable amount of chip area.

Figure 2.6 – Charge Redistribution DAC with Offset Cancellation
**Hybrid Architectures**

Hybrid DACs comprise of the combination of the architectures described in previous sections at different portions of the DAC, [2] and [6]. The hybrid approach is very popular as it, if properly designed, combines the advantages of different architectures. The most popular hybrid architecture is the combination of a binary-weighted architecture and a thermometer-coded architecture. As this thesis's focus is on the advantages of decoding schemes and segmentation, the details of the advantages of the hybrid or segmented architectures will be discussed more in depth in the following chapter.
Overview

In general, three decoding schemes are used resulting in three DAC architectures: binary-weighted, thermometer-coded, and segmented DACs. Decoding schemes have a significant effect on both static and dynamic performance.

Binary-weighted Architectures

As no decoding logic is required, a binary-weighted DAC has higher conversion speed, offers simplicity in routing and has low power consumption. However, this structure has major drawbacks on both static and dynamic performance. These drawbacks are all associated with major transitions of the DAC and the severity of the problem is proportional to the weight of the bit. The worst case occurs at the middle-code transition when the MSB current source of the binary-weighted array is being turned on and all the other current sources are being turned off. Therefore, this architecture is not guaranteed to be monotonic and may result in large DNL error.
Thermometer-coded Architectures

In thermometer decoding, a digital decoder is required to convert the binary input values to a thermometer-coded equivalent. In thermometer coding, one more output on the decoder is selected for every increase in the binary input code. Therefore, advantages of thermometer-coded based architectures are that they offer guaranteed monotonic outputs, low DNL errors and reduced glitch noise. The major drawback of thermometer-code based architectures is that it requires a decoder which draws extra power and takes additional area to implement. To represent $2^n$ different digital values, the thermometer code requires $2^n-1$ levels. Therefore, for higher $n$ values, the decoding logic can introduce large decoding delay and could take a lot of real estate on silicon. Also, for high speed DAC design, the decoder design is not trivial.

Segmented Architectures

The widely used segmented DACs utilize the advantages of binary-weighted and thermometer-coded architectures. Generally, certain number of most significant bits is implemented in thermometer-coded architecture and the rest of the least significant bits are realized in binary-weighted fashion. In doing so, the low DNL of thermometer-coded
architecture can be exploited to achieve better overall DNL performance while reducing the decoder logic and area by using binary-weighted architecture in places where the overall DNL is least affected.

**DNL Performance of Decoding Schemes**

For the current steering DACs shown in Figure 3.1 and 3.2, we can express the current of the LSB when there is only random mismatch and if the gradient and finite output impedance effects are neglected as,

\[ I_j = I \times (1 + \epsilon_j), \quad j = 1,2,3,...,2^n - 1 \]  

(3.1)

where ‘n’ is the resolution of the DAC and \( \epsilon_j \) is the error at the \( j^{\text{th}} \) current source. DNL at the digital input code \( D \) can be expressed as [7],

\[ DNL(D) = \frac{I(D) - I(D-1)}{\bar{I}} - 1 \]  

(3.2)

where

\[ \bar{I} = \frac{I(N) - I(0)}{N} \]  

(3.3)

which is the average current and \( N = 2^n - 1 \).

In addition to the matching characteristics of the components, the DNL performance of the system depends on the decoding method employed. In a binary decoded architecture, the maximum DNL error occurs at mid code transition when the MSB current source of the binary-weighted array needs to match the sum of all the other current sources. Therefore, by using (3.1) and (3.2), we can calculate DNL at this point as,
If the mismatches between the unit current sources are random and uncorrelated, and each current source has the same relative standard deviation \( \sigma \), then the standard deviation of DNL at this transition is approximately equal to,

\[
\sigma_{DNL}(midcode) = \sigma \cdot \sqrt{\left( \frac{2^n - 2}{2^n - 1} \right)^2 \cdot \left( \frac{2^n}{2^n - 1} \right) \cdot \left( \frac{2^n}{2^n - 1} \right) \cdot \left( \frac{2^{n-1}}{2^{n-1} - 1} \right)}
\]

If \( N' = 2^n \) then,

\[
\sigma_{DNL}(midcode) = \sigma \cdot \sqrt{\left( \frac{N' - 2}{N' - 1} \right)^2 \cdot \left( \frac{N'}{N' - 1} \right) \cdot \left( \frac{N'}{N' - 1} \right) \cdot \left( \frac{N'}{2} - 1 \right)}
\]

\[
= \sigma \cdot \sqrt{\left( \frac{N' - 2}{N' - 1} \right) \cdot N'}
\]

\[
= \sigma \cdot \sqrt{\frac{N'^2 - 1}{N}} = \sigma \cdot \sqrt{N - \frac{1}{N}} \quad (3.4)
\]

For a thermometer decoded architecture,

\[
DNL(midcode) = \frac{I(100\cdots0) - I(011\cdots1)}{I} - 1
\]
The standard deviation in this case is approximately equal to,

\[
\sigma_{DNL\ (midcode)} = \sigma \cdot \sqrt{\left( \frac{n-2}{2^n-1} \right)^2 + \frac{n-2}{(2^n-1)^2}}
\]

\[
= \sigma \cdot \sqrt{\left( \frac{N-1}{N} \right)^2 + \frac{N-1}{(N)^2}}
\]

\[
= \sigma \cdot \sqrt{\frac{N-1}{N} = \sigma \sqrt{1 - \frac{1}{N}}}
\]  \hspace{1cm} (3.4)

However, in a segmented architectures, where lower ‘n1’ bits are realized in binary-weighted structures and upper ‘n2’ bits are decoded in thermometer-coded segments and n2 + n1 is the total resolution ‘n’ bits of the overall architecture, the overall DNL of the n2 + n1 segmented DAC can be expressed as,

\[
\max(\sigma_{DNL\ segmented}) = \sigma \sqrt{(n1+1) - \frac{1}{(n1+1)}}
\]  \hspace{1cm} (3.6)

By using (3.4), (3.5) and (3.6) we can calculate the worst case DNL for the different decoding schemes in terms of number of bits, which is shown in Figure 3.3. In segmented DAC case, the n1 is the number bits in the binary-weighted lower bits.
Yields for Different Decoding Schemes

In designing a DAC, it is crucial to consider the INL and DNL yields. Considering the same current steering DAC shown in Figure 3.1, random mismatches in the components will also affect the INL and DNL yields, [16]. The INL or DNL yield can be determined as the percentage of total DACs which meet or exceed the specific INL or DNL level. It can be shown that decoding schemes can greatly impact on the INL and DNL yield.

![Graph showing comparison between DNL performances of different number of bits in binary-weighted and thermometer-coded architectures.](image)

Figure 3.3 – Comparison between DNL performances of different number of bits in binary-weighted and thermometer-coded architectures

To meet a specific DNL or INL performance, for example ±A LSB, the magnitude of the INL or DNL at any given digital code must not be larger than A LSB. The INL or DNL yield, for a normalized DAC, can be described as,

\[ \phi(n,A) = P[|INL(D)| \leq A, D = 1,2,\ldots,N-1] \]
If the relative standard deviation of each unit current source is \( \sigma \), the INL (or DNL) yield of the DAC is equal to,

\[
Y = \phi \left( n, \frac{A}{\sigma} \right)
\]

(3.8)

The above expression shows the relationship between the minimum required standard deviation of unit current sources (\( \sigma \)), the bits of resolution (\( n \)), the INL (or DNL) specification (\( A \)) and the corresponding yield (\( Y \)).

When \( n \) is large and \( \sigma \) is small, it is shown in [7] that the minimum required standard deviation of the unit current sources can be expressed as,

\[
\sigma \approx \frac{A}{\sqrt{2^Y}} Z(Y)
\]

(3.9)

\( Z(Y) \) is dependent only on yield requirement and is determined for several number of bits with \( A=0.5 \) using Monte Carlos simulation in [7] as shown in Figure 3.4 and 3.5.

![Figure 3.4 – \( Z(Y) \) INL yield for thermometer-coded DACs](image)
As the INL and DNL yields for the thermometer-coded DACs are the same, and the INL yield for binary-weighted DACs are equivalent to that of thermometer-coded ones, we will use Figure 3.4 to calculate DNL and INL requirement for thermometer-coded DACs and INL requirements for binary-weighted DACs. The DNL yield for binary-weighted DACs will be calculated using Figure 3.5.

![Graph](image)

**Figure 3.5 – Z(Y) DNL yield for binary-weighted DACs**

Using (3.9), we can determine the minimum $\sigma$ requirement of ‘n’ bit architecture for specific yield level. For fully thermometer-coded n bit architecture with DNL yield of 99% and DNL of $\pm0.5$ LSB, for example, the minimum $\sigma$ required is 10.6%. We will use this $\sigma$ as the base to calculate the yield of different number of bits with different architectures for DNL of $\pm0.5$ LSB. Figure 3.6 shows comparison among the yields of different resolutions of binary-weighted, thermometer-coded and segmented DACs. The yield in terms of the number of bits in lower binary-weighted LSB bits is showed for segmented DACs.
Digital and Analog Area Requirements for Different Coding Schemes

As we observed in Figure 3.6, the yield is greatly affected in the binary-weighted and segmented architecture for a given $\sigma$ requirement. However, we can improve the yield for both types of architectures by reducing the $\sigma$ requirement. As the $\sigma$ of a current source is dependent on the area of the current source as,

$$\text{Area} \propto \frac{1}{\sigma^2}$$  \hfill (3.10)

we can decrease the $\sigma$ by increasing the area of the current source.

![Figure 3.6 - DNL (Yield) for different coding schemes](image)

Again, we will use the area of a fully thermometer-coded DAC with a DNL yield of 99% and DNL of $\pm0.5$ LSB as the base unit area. Then, we calculated the area necessary to
get the same performance and yield for binary and segmented structures. Figure 3.7 shows the results of the calculations.

Also, in both segmented and thermometer-coded architectures, the complexity and the area of the digital logic for different number of bits increase exponentially. Figure 3.8 shows the relative complexity of thermometer-coded decoder normalized over the logic for binary-weighted structure of same resolution and the segmented decoder normalized over the logic for a 14-bit binary-weighted architecture.

![Figure 3.7 - Area requirements for binary and segmented architectures](image)

**Conclusion**

It can be clearly seen that all the decoding schemes have both advantages and disadvantages in terms of DNL and yield performance. The DNL and yield performance of thermometer-coded structure is excellent compare to other architectures but the logic requirement can get very complex at high resolution DACs. Therefore, the popularity of
segmented architectures in the literature can be justified due to the overall DNL and yield performance with reasonable amount of complexity involved in the decoder logic.

Figure 3.8 – Relative digital logic complexity for thermometer-coded and segmented architectures
CHAPTER 4

OPTIMIZATIONS TECHNIQUES FOR DECODING LOGIC

Overview

As discussed in the previous chapter, segmentation can offer benefits in terms of DNL and yield performance of a DAC. However, for high resolution designs, the segmentation of a large number of the MSBs is required to achieve low DNL and to minimize the current source area while maintaining a desired yield level. In this case, the complexity in the decoding logic becomes a major drawback in this type of architecture. This generally results in longer delays in the digital decoder and this, in turn, adversely affects the speed and induces a large amount of silicon real estate for the logic circuit. There have been efforts to relax the complexity and the size of the digital decoder logic and several of them will be discussed in this chapter.

Row/Column Decoding Logic

In [8] and [9], the decoder for the MSB thermometer-coded structure is divided into a row and a column decoder so that the logic in the decoder is greatly reduced. In this approach, a small and simple logic circuit at an individual current source decides whether the current source has to be turned on or not based upon signals generated from the row and column decoders. The overall block diagram of the decoding logic for a row/column decoder is shown in Figure 4.1.

The decoding logic is implemented in two steps. The digital inputs are first decoded into thermometer-coded signals used to drive the row and column lines. Then the logic
circuit in each individual cell determines whether to turn the current source at a given position on or off at the corresponding row and column position. In doing so, the decoding logic is substantially simplified resulting in lower real estate on silicon and faster decoding speed. There is a modest overhead in terms of routing the extra row and column signals into the cell matrix and in including the simple cell-level logic. However, in today’s standard processes which include multiple layers of metal with the ability to stack vias on top of each other, the routing overhead is minimal.

![3 UMSB bits](image)

Figure 4.1 – Row/Column Segmentation Implementation

This technique is attractive for both fully thermometer-coded structures and segmented architectures where the boundary of segmentation can be adjusted to include more thermometer-coded LSB bits. By doing so, the area requirement for obtaining a given DNL yield can be reduced. However, this row/column two dimensional decoder still requires
considerable decoding logic circuitry when used for a very high speed DACs with more than 10 bits of resolution in the thermometer-coded part of the structure [10]. This decoding logic requirement creates a bottle neck for high-speed high-resolution applications.

Nested Row/Column Decoding Logic

In order to further capitalize on the advantage of the “two-dimensional” row/column decoder concept, we propose to extend the idea and implement a multi-dimensional nested row/column decoding strategy to achieve high speed decoding at high resolution. A similar idea has been realized in [10], but the design was more geared towards the combination of several blocks of logic proposed in [9]. The nested row/column decoding could make it possible for the designer to implement fully thermometer-coded architectures at high resolution and high speed or segmented structures with enough thermometer-coded bits to obtain good DNL yield at high resolution levels.

The multi-dimension nested row/column decoding logic proposed consists of multiple levels of matrix decomposition based on row/column logic. In what follows I will describe only two-level nesting. The first stage in a two-level nested decoder consists of matrices of current cells controlled by a single common row/column decoder which is similar to the structure shown in Figure 4.1. The second stage is comprised of a single matrix comprised of the first stage matrices. The second state is controlled by a separate row/column decoder. The top level block diagram of a 12-bit design is shown in Figure 4.2.
The decoder logic is implemented as follows. The binary input bits are divided into two halves: the first-half binary bits (FH) comprised of the LSB bits and second-half binary bits (SH) comprised of the MSB bits. The first half binary bits are then divided further into two groups: lower first-half binary bits (LFH) and the upper first-half binary bits (UFH). The LFH and UFH bits are used to control the selection of current cells in a single matrix. The second-half binary bits are also divided into two groups: the lower second-half binary bits (LSH) and the upper second-half binary bits (USH). The LSH and USH bits are used to decide which matrices need to be turned on.
X = Next UFH bit, Y = Current UFH bit, Z = Current LFH bit

\[ P = \text{Next USH bit}, \quad Q = \text{Next LSH bit} \]

Figure 4.3 – Current Cell with Decoder Logic Circuit

All four control signals, LFH, UFH, LSH and USH, are decoded using 3-to-8 decoders with the standard truth table shown in Table 4.1. All decoded signals are feed into every single current cell in the matrix. The decoder logic circuit at the current cell then decides whether to turn on or off the corresponding current source based upon these signals. The logic can be partitioned into two parts: one at the matrix level and the other at the current cell level. At the matrix level, the decoded LSH and USH signals are used to differentiate matrices into 3 different types. They are: (1) matrices where all current cells in them are turned on, (2) matrices where all current cells in them are turned off, and (3) a single matrix where current cells are turned on or off based on cell level logic.

As shown in Figure 4.3, when either the next column or next row signal is logic high, all of the cells in the current matrix are set to turn on. If both the next column and next row signals are logic low, the cells in the matrix are turned on according to logic from the cell level decoder. If both of the cases are false, all of the current cells in the matrix are turned off. In the cell level logic, the same decoder circuit proposed in [8] is used.
The nested row/column decoding logic offers two advantages: first, it can be used to move the boundary of segmentation to more least significant bits so that smaller areas can be used to get the necessary DNL yield performance and second, the decoding logic circuit becomes much more efficient at higher bit levels thus offering improvements in high speed applications. However, if we look at the total numbers of gates required to realize both the row/column decoder and nested row/column decoders, there is an increase in overall gate count. This increased gate count is due to the simple logic circuits needed at every cell in the matrix. Figure 4.4 shows the comparison between gate counts for 8-bit conventional binary-to-thermometer, row/column, and nested row/column decoders.

The advantage of nested row/column decoder can be more evident at the higher number of bits. The comparison between a 12-bit row/column decoder and similar nested row/column decoder is shown in Figure 4.5. It can seen that the nested row/column decoder is nearly 4 times as fast as the row/column decoder but takes about 1.9 times the number of gates.
Linear-step Decoding Scheme

In the previous sections, it is shown that thermometer-coded architectures have better DNL yield than binary-coded approaches. A comparison between the combinational logic decoder and row/column decoder to implement the binary-to-thermometer decoding is also made. To increase the speed of the decoding circuit, a nested row/column decoding scheme which is based on row/column decoding scheme is also proposed. However, the total number of digital logic gates required to implement either the row/column or nested row/column decoder increases with the number of bits in resolution. In the following sections, we will investigate decoding schemes to lower the total number of digital gates used for the decoding logic.
In order to examine a possible way to reduce the requirements for the digital decoder, the relationship between the number of current outputs and the number of gates in the decoder circuit is analyzed. The number of current outputs in the data converter will be defined as the total number of switches with varying number of current values. For an n-bit architecture, the total number of outputs \( N \) is,

\[
N = \left(2^n - 1\right)
\]  

(4.1)

We can expand (4.1) as,

\[
N = \left(2^n - 1\right) = 2^0 + \sum_{i=1}^{k} 2^{m_i}
\]  

(4.2.1)

The factors \( k \) and \( m \) depend on the decoding scheme. For fully thermometer-coded structure,

\[
k = 2^n - 1 \quad \text{and} \quad m = 0
\]  

(4.2.2)
For fully binary-weighted approach,

\[ k = n - 1 \quad \text{and} \quad m = 1 \quad \text{(4.2.3)} \]

![Diagram showing the relation between total number of output switches and number of gates for different bits of resolution in thermometer-coded architecture.](image)

**Figure 4.6 – Relation between total number of gates and number of outputs**

The total number of output switches in either architecture is equal to \( k+1 \). To obtain the gate count, gate level logic circuit is generated from Verilog code by using Synopsys circuit extractor. Figure 4.6 shows the correlation between the number of output switches and the number of gates for different bits of resolution in thermometer-coded architecture. It can be observed that number of gates is directly related to the number of output switches present. It can also be shown that the speed of the decoder is also correlated to \( k \). Figure 4.8 shows that, as \( k \) gets larger, the number of gates in the critical path becomes large and hence, decoding speed becomes slower.
Therefore, we can effectively reduce the complexity of the combinational decoding logic, and consequently the decoding speed, by lowering the number of output switches used in the circuit. In order to do so, we expand (4.2.1) in a different way as,

\begin{align*}
N &= (2^n - 1) = 2^6 + 2^4 \left[ 1 + 2 + 3 + \ldots + i \right] \\
&= 2^6 + 2^4 \sum_{i=1}^{k} i \cdot \alpha_i = 1 + \delta \cdot \sum_{i=1}^{k} i \cdot \alpha_i, \quad \text{where } \alpha_i \in \{0,1\} \quad (4.3)
\end{align*}

Figure 4.7 Comparison of outputs in three different coding schemes

The value \( k \) and \( \alpha_i \) depend on \( N \). The term \( \delta \) will be defined as the step value. By carefully choosing proper values for \( k, \alpha_i \) and \( \delta \), any output between 0 and \( N \) can be realized. Figure 4.7 shows the comparison among the number of resolution, \( n \), and the total number of outputs in binary-weighted coding, thermometer coding and linear-step coding with \( \delta = 2 \). (4.3) can be interpreted as a thermometer-coded expansion of \( N \) with linearly...
increasing $\delta$ steps. We can prove that the higher number of outputs generally results in a longer delay in the decoder circuit. Figure 4.8 shows the relation between the resolution of the data converter and the total number of gates in the longest signal path.

![Graph showing relation between resolution and number of gates](image)

Figure 4.8 – Number of outputs vs. gates in critical path

The switching sequence in this case is similar to that of a binary-weighted decoder. The lower values bits are turned off when a higher value bit is being turned on. However, the DNL error in this case is smaller than that of a binary-weighted fashion since the difference between the smaller bit values and the highest bit value is smaller than that of in binary-weighted scheme. We can calculate the standard deviation in DNL of linear-step decoding by using (3.4). Doing so, we will plot the sigma of linear-step decoding scheme against those of binary-weighted and thermometer coded schemes as shown in Figure 4.9.
Multi-segmented Linear-step Decoding Scheme

We will further try to reduce the DNL error in the linear-step decoding scheme by changing the length of occurrence of each $\delta$ step in (4.3). Doing so, we will express (4.3) as,

$$N = (2^n - 1) = 2^0 + 2^1 \cdot [1 + 2 + 3 + ... + i]$$

$$= 2^0 + 2^1 \cdot [1 + 1 + 1 + ...] + 2^1 \cdot [2 + 2 + 2 + ...] + ... + 2^1 [i + i + i + ...]$$

$$= 1 + \delta \cdot \sum_{i=1}^{l} \sum_{j=1}^{k} j \cdot \alpha_{ij}, \alpha_{ij} \in \{0,1\} \quad (4.3.1)$$

We can further modify (4.3) to obtain the $\delta$ value of 4. Then it becomes

$$N = 1 + 2 + 4 \sum_{i=1}^{l} \sum_{j=1}^{k} j \cdot \alpha_{ij}, \alpha_{ij} \in \{0,1\} \quad (4.3.2)$$
Using (4.3.1) we can prove that when $l$ is increased, the DNL error is reduced but the number of outputs increased. When $\delta$ is increased, the DNL error is increased but the number of outputs reduced. By choosing proper $l$ and $\delta$ values, a switching sequence with the optimal level of both DNL error and the number of outputs, which means reduced logic requirement, can be obtained. Figure 4.10 shows the DNL error comparison between DNL error among linear-step schemes with $\delta = l = 1$, $\delta = l = 4$, and thermometer coded structure.

![Figure 4.10 - Effect of $l$ and $\delta$ on DNL error](image)

Choosing $\delta = l = 4$, Figure 4.11 shows the comparison of number of output switches between the fully thermometer-coded, fully binary weighted and the multi-segmented linear step decoding structures. It can be clearly seen that the multi-segmented linear-step decoding scheme can lower the number output switches associated. Also, the number of gates required
to realize the structures and the number of gates in the critical path are compared in Figure 4.12 and 4.13 respectively.

![Figure 4.11 - Number of output switches in different decoding schemes](image)

**Conclusion**

As can be seen in the comparisons, the multi-segmented linear-step decoding scheme can reduce the number of output switches and thus lowering the circuit requirement and gate delays in the critical path. However, one can examine that due to linearly increasing $\delta$, the effective DNL performance of the multi-segmented linear-step decoded DAC will be higher than that of a fully thermometer-coded one, although it will be better than a fully binary-weighted architecture.
Figure 4.12 – Gate count comparison among different decoding schemes

Figure 4.13 – Gates in the critical path in different decoding methods
CHAPTER 5

A 10-BIT DIGITAL-TO-ANALOG CONVERTER USING MULTI-SEGMENTED LINEAR-STEP DECODER

Overview

In the previous chapter, various methods of reducing the logic requirement for the decoder circuits were discussed. It was shown that using multi-segmented linear-step decoding logic can lower the number of gates required. In order to verify the effectiveness of the method, a 10-bit DAC has been designed and fabricated in a 0.6 micron technology.

Architecture Selection

A current steering approach is selected to realize the DAC as it does not require an output buffer which can affect the overall speed of the structure. The top-level block diagram is shown in Figure 5.1. By choosing \( l = \delta = 4 \), a total of 46 output switches are necessary in the output switch block. The current cell block includes 1059 unity current cells.

Analog Design Aspects

Power Consumption

The dominant part of power consumption lies in the analog current cell part of the digital-to-analog converter. As the output switches are differential, the power is being consumed by all the cells at anytime regardless of the input value. Therefore, the total power consumption of the architecture can be estimated from the analog part as the power...
consumption at the digital decoder is usually negligible due to dynamic nature of the CMOS logic cells. The total power consumption can be estimated as,

\[ P_{\text{total}} \approx \text{total number of current cells} \cdot I_{\text{ref}} \cdot V_{\text{dd}} \quad (5.1) \]

In this design, \( V_{\text{dd}} \) of 5V is selected and therefore, it is necessary to minimize the total current in the design to lower the total power consumption. A total current value of 10 mA takes about 50 mW of power.

Figure 5.1 – Top level block diagram of 10-bit DAC
Matching Requirement

In order to achieve the desired current output from a CMOS transistor in saturation, neglecting the channel length modulation factor, the following SPICE Level 1 equation can be used.

\[ I_D = \mu C_{ox} \frac{W}{2L} (V_{GS} - V_T)^2 \]  

(5.2)

As the gate bias voltage, \( V_{GS} \), can be fixed as a constant and the transconductance parameter, \( \mu C_{ox} \), and the threshold voltage, \( V_T \), are considered as constant values for a specific process, the width, \( W \) and the length, \( L \) are varied to achieve desired current value. It can be noted that to minimize the area requirement for the current cells, it would be an ideal case to make the minimum width and length dimensions. However, due to random mismatches in the process parameters, the current sources have to be designed to offset the mismatch errors.

Based on Pelgrom’s model [11], two identically designed MOSFET in saturation with a certain overdrive voltage of \( V_{GS} - V_T \), the standard deviation is described as,

\[ \frac{\sigma^2(I_D)}{I_D^2} = \frac{4 \cdot \sigma^2(V_{T0})}{(V_{GS} - V_T)^2} + \frac{\sigma^2(\beta)}{\beta^2} \]  

(5.3)

where,

\[ \sigma^2(V_T) = \frac{A_{VT}^2}{W \cdot L} + S_{VT}^2 D^2 \]  

(5.3.1)

and

\[ \frac{\sigma^2(\beta)}{\beta^2} = \frac{A_{\beta}^2}{W \cdot L} + S_{\beta}^2 D^2 \]  

(5.3.2)
$A_{r_T}$ and $A_p$ are process dependent constants.

By using (3.9) and Figure (3.4), we can obtain the minimum $\sigma$ required to achieve the specified yield and INL for n bit resolution. In this case, for 10-bit architecture with $\pm 0.5$ LSB of DNL for 99% yield, the minimum $\sigma$ is 0.93%. For $A_{r_T}$ and $A_p$ for AMI 0.5 micron process and using (5.3), (5.3.1) and (5.3.2), the minimum gate area required to achieve the matching performance for the above specification is $9.12 \ \mu m^2$.

**Gradient Errors**

A wide range of aspects contribute to the gradient errors. Doping and oxide thickness variations across the wafer or the voltage drop along the power lines can cause approximately linear gradient errors [12]. Temperature gradients and die stress may introduce approximately quadratic gradients [13]. As gradient effects tend to introduce serious systematic errors in large structures, various switching techniques have been proposed in [7]. In order to optimize the signal routing involved with the multi-segmented linear-step decoding, a two-level hierarchy switching sequence is used in this work. For $\delta = 4$ and $l = 4$, (4.3.2) is used to get optimal number of current cells to achieve 10-bit of resolution as follows. As the current values are changed with every 4 steps, the current cells are divided into 4 first level quadrants as shown Figure 5.2. Also, due to the fact that the current difference is every step is also 4 $I_{ref}$, a total of 4 first level quadrants are combined to make a second quadrant as shown in Figure 5.3. Finally, a total of 4 second level quadrants are used to achieve the final layout of the structure shown in Figure 5.4.
Figure 5.2 – Layout of a first level current cells quadrant

Figure 5.3 – Layout of a second level current cell quadrant

Figure 5.4 – Layout of the final current matrix
Output Impedance

After matching and gradient error considerations, the finite output impedance of each current source is also another factor that can cause errors in the DAC output, [14]. It has been shown that the requirement for output impedance is not trivial at high resolution, [7]. One of the methods to relax this strict requirement is to use differential outputs with the expense of significantly more complex routing requirements. However, even with the differential outputs, achieving the level of output impedance for a high resolution DAC is still a challenge with a single transistor. Therefore, a cascode stage can be added to effectively increase the output impedance, [15]. However, the adding of a cascode stage can limit the output signal swing. The current cell design used in the prototype is shown in Figure 5.5. It contains a single current cell with a pair of switches driven by a switch driver.

![Figure 5.5 - Current Cell](image)

The switch driver, Figure 5.6, limits the signal swing on the output node A and therefore, reduces glitches and increase the settling speed. It also keeps the switches in saturation region making them as cascode stage for the current cell, and thus increasing the
output impedance. The latch synchronizes the digital signals to minimize the timing skew associated with different routing delays.

![Switch Driver Circuit](image)

**Figure 5.6 – Switch Driver and Latch**

**Digital Design Aspects**

The decoder logic is implemented with Verilog Hardware Description Language (Verilog HDL) using behavioral modeling. This model is used for verification simulations along with transistor level circuit for analog portion. After verification, the Verilog model is extracted with Synopsys to obtain the gate level circuit and description. Then, the layout of the decoder circuit is generated using standard cells and gate level description.

**Simulation and Testing Results**

**Prototyping and Simulations**
The circuit was first simulated in schematic level without parasitic components using Spectre to verify the functionality and specifications. After successful schematic level verification, the structure is put into layout and extracted with parasitic components. The extracted circuit is then simulated again with linearly varying transistor size matching to verify the results obtained previously. Figure 5.7 to Figure 5.10 show the simulation results. They are obtained with a 10 MHz clock signal with Nyquist rate data signals.
Figure 5.9 – INL plot from extracted circuit simulation

Figure 5.10 – DNL plot from extracted circuit simulation

**Experimental Results**

A prototype circuit was fabricated through MOSIS Education Program using the AMI 0.5 micron 3-metal/2-poly process. The chip micrograph is shown in Figure 5.11. The chip requires a total area of 0.63 mm$^2$. It can be seen that the digital portion of the design takes about one-third of the layout. This is due to limited availability of standard cells, and the digital decoder could not be optimized very efficiently.
Static Performance

The DNL of the DAC is tested as follows. Digital code $k$ is put in and the corresponding analog voltage output $V_o(k)$ is measured. Then, digital code $k+1$ is put into the DAC and the corresponding analog voltage output $V_{\text{out}}(k+1)$ is measured. The digital code $k$ is then put in again and the analog output $V_o(k)'$ is measured. If $V_o(k) = V_o(k)'$, then

$$DNL_k = \frac{V_o(k) - V_o(k + 1)}{V_{\text{average}}} - 1 \quad (5.4)$$
If $V_o(k) \neq V_o'(k)$, then the whole process is repeated until the condition is achieved. By doing so, the DNL of all digital codes between 0 and N is determined. Figure 5.11 shows the DNL performance. The highest DNL is $\pm 0.52$ LSB. In the process, the DUT temperature is maintained at $25 \pm 0.1^\circ C$ using active feedback air thermal stream to minimize thermal variation during the measurement process.

![DNL performance graph](image)

Figure 5.12 – DNL performance

The INL of the DAC is also measure as follows. First, $V_o(k = 0)$, the output voltage at digital code $k=0$ and $V_o(k = N)$, the output voltage at digital code $k=N$ are measured. Then, $V_o(k)$, the output voltage at digital code $k$ between 0 and N is measured. The fit line across the output at digital code $k$ is then obtained. The output voltages at digital codes $k=0$ and $k=N$ are measured again to obtain $V_o(k = 0)'$ and $V_o(k = N)'$. Then, the slope and the
offset of the fit line across the output code \( k \) is checked using the new values at \( k=0 \) and \( k=N \) to ensure it is the same as before. If either parameter has changed, the process is repeated. Otherwise,

\[
INL_k = V_o(k) - V_o(k)'
\]  

(5.5)

where \( V_o(k)' \) is the fit line output evaluated at code \( k \). Figure 5.13 shows the plot of INL error across all input codes. The highest INL error is \(-1.02/0.55 \) LSB. Again, the DUT temperature is maintained at \( 25 \pm 0.1^\circ C \).

![Figure 5.13 – INL performance](image)

**Dynamic Performance**

The spectral performance of the DAC is determined using FFT method in MATLAB. Table 5.1 shows the results obtained. Figure 5.14 shows the output spectrum of the DAC.
Table 5.1 – Spectral performance data

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>SFDR</td>
<td>55.92 dB</td>
</tr>
<tr>
<td>ENOB (SFDR)</td>
<td>9.33 bits</td>
</tr>
<tr>
<td>THD</td>
<td>55.43 dB</td>
</tr>
<tr>
<td>ENOB (THD)</td>
<td>8.91 bits</td>
</tr>
</tbody>
</table>

Figure 5.14 – Power spectrum of output

Conclusions

As digital-to-analog converters are becoming more and more important with various signal processing and telecommunication systems, improvements in resolution and speed are becoming more critical. The proposed nested row/column decoding scheme can reduce the delay in the decoding logic thus offering improvements in speed of operation at the expense
of a modest increase in the number of gates. Multi-segmented linear-step decoding logic can also be used to reduce the total number of gates and the time delay to moderate levels. A 10-bit current steering DAC was implemented using the multi-segmented linear-step decoding logic to validate this approach. It was fabricated in 0.5 micron 3-metal/2-poly process and provided anticipated performance at the 10-bit level however the number of samples available was not adequate to validate DNL yield potential of the circuit.
REFERENCES


ACKNOWLEDGEMENTS

It has been a long and hard-fought journey for me to come to this moment. Born in Burma, which is one of the least developed nations of the world, I feel a great honor to be able to obtain a graduate degree in one of the most advanced fields in engineering. Of course, every flying wings have wind beneath and I shall take this opportunity to reflect and acknowledge their support.

The first person to mention here is Dr. Randall Geiger, who has been my advisor since I was an undergraduate student at Iowa State University. His extraordinary knowledge on circuits and systems has been a model and has always provided guidance whenever I needed. Also, his patience and his support towards all of his students gave me a rewarding experience during my years as a graduate student. I could not mention enough here for having a chance to learn from him.

I also owe a lot of appreciation towards Greg Dace and Joel Hagen from Acumen Instruments Corporation for their sustained support and guidance. My time with them as an intern turned out to be one of the most valuable and enjoyable times during my academic journey.

Need we say more that your colleagues are crucial part of your work and your success? Without their support, it would not have been a pleasant journey. I must mention Vipul Katyal, Haibo Fei, Ahmed Hashim, Lance Juffer, Beatriz Olleta, Eddie Boylston, Su Chao, Yu Lin, Saqib Malik, and Mark Schlarmann for their support. Most of all, I have to single out Vipul for his amazing wit on math problems, Haibo for his help with simulation tools and circuits, Ahmed for being the “Master Jedi”, and Beatriz for her charming support.
Also, I would like to thank Saqib and Mark for their help with Cadence and circuits. In addition, I have to point out Brian Reed, Rius Tanadi, Sven Soel, Imad Abbadi, Mark Taylor, Tyson Benson, Seth Hendrickson, and Luping Zhou for their friendship and support throughout the time.

I’d also like to thank Dr. Degang Chen and Dr. Stuart Birrell for their support and help as part of my program of study committee. Also, I have to mention Dr. Richard Englehorn, Dr. Ken Kruempel, Dr. Zhao Zhang for their guidance and help.

Last but not least, I would like to thank my family for their continued support and encouragement. I would like to thank Johnson Lim for his continued support. Especially, I have to mention my wife, Vivian Wong, for her love, support and patience during the time apart. Finally, I would like to thank my mother, Khin Hnin Aye, for supporting my academic endeavors since from day-one. Without her unconditional love, support, guidance, and understanding, I would not be here where I stand today. For these, I dedicate this work to my family.