The Thesis Committee for Li You
Certifies that this is the approved version of the following thesis:

Design and Implementation on High-order Mismatch-shaped Multibit Delta-Sigma D/A Converters

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

Nan Sun

David Z. Pan

Design and Implementation on High-order Mismatch-shaped Multibit Delta-Sigma D/A Converters

by

Li You, B. S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ENGINEERING

The University of Texas at Austin

May 2014

Dedicated to my parents

Acknowledgements

I have been very fortunate and honored to have Professor Nan Sun as my advisor during my graduate study at the University of Texas at Austin. He is not only an outstanding mentor in academic work, but also inspires me with his dedication to research. I greatly appreciate all the things that I have learned from him, which will benefit me all along my life.

I express my deep gratitude to Professor David Z. Pan to be the member of the committee. I would like to thank all the professors who taught me during my two-year study at University of Texas at Austin.

My appreciation also goes to my friends at the University of Texas at Austin for their great support, continuous encouragement, insightful discussion, and generous sharing: Boyang Zhang, Long Chen, Xiyuan Tang, Jaeyoung Park, Peijun Wang, Ji Ma.

I dedicate my thesis to my parents in China. Without their love, encouragement and sacrifice, I would not have been as I am.

Abstract

Design and Implementation on High-order Mismatch-shaped Multibit Delta-Sigma D/A Converters

Li You, M.S.E

The University of Texas at Austin, 2014

Supervisor: Nan Sun

As the rapid evolution in semiconductor technology, transistors' feature size has reached to 22nm and below, which brings great impact to analog and mixed-signal circuits. As the significant bridge connecting the analog world and digital system, data converter suffers from nonlinearity resulting from mismatch among its unit components. The smaller transistors are, the larger relative mismatch among them becomes. However, using larger transistors leads to more area cost and power consumption. Therefore, researchers have been working hard on how to alleviate the mismatch issue. In recent years, Dynamic Element Matching (DEM) becomes a popular approach that can significantly improve linearity, especially Spurious-free Dynamic Range (SFDR), of a data converter system. The basic idea of DEM is to shuffle the usage pattern of unit elements so that the mismatch error is no longer correlated to the input signal. Thus, DAC's linearity will be improved. Generally, DEM Nyquist-rate DAC does mismatch scrambling, which smooths distortions resulting from mismatch into white noise. DEM Delta-Sigma DAC does mismatch shaping, which pushes distortions away from the signal band, typically lower frequencies.

In this thesis, we focused on mismatch-shaping Delta-Sigma DACs. Two of those various algorithms are implemented logically and physically. With placement and routing information, we got more accurate result on the speed and power dissipation. The comparison shows the tradeoff among number of quantization levels, mismatch-shaping order, and hardware complexity.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Generally, a multibit Digital-to-Analog Converter (DAC) has several elements, each of which takes one bit from DAC's input. All elements' outputs are summed up to generate the final output of the DAC. The various DACs are essentially different in how input bits are delivered to the elements and how outputs from the elements are weighted and summed up to form the final output [1]. For example, binary, thermometer and segmented DACs are some of the traditional DAC structures.

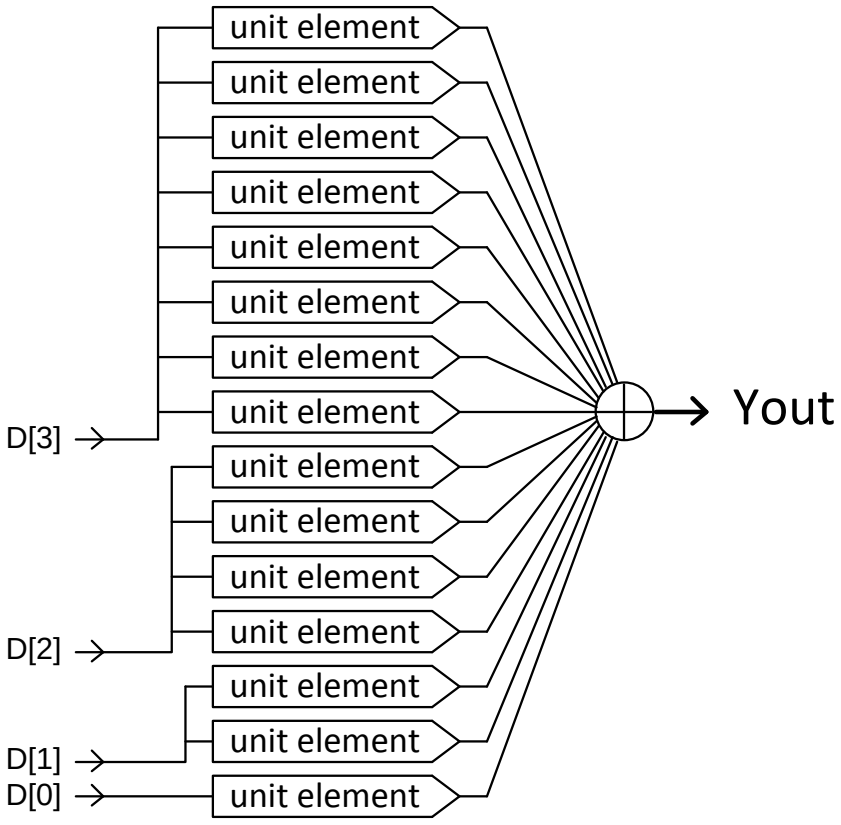


Fig. 1.1 4-bit binary DAC block diagram

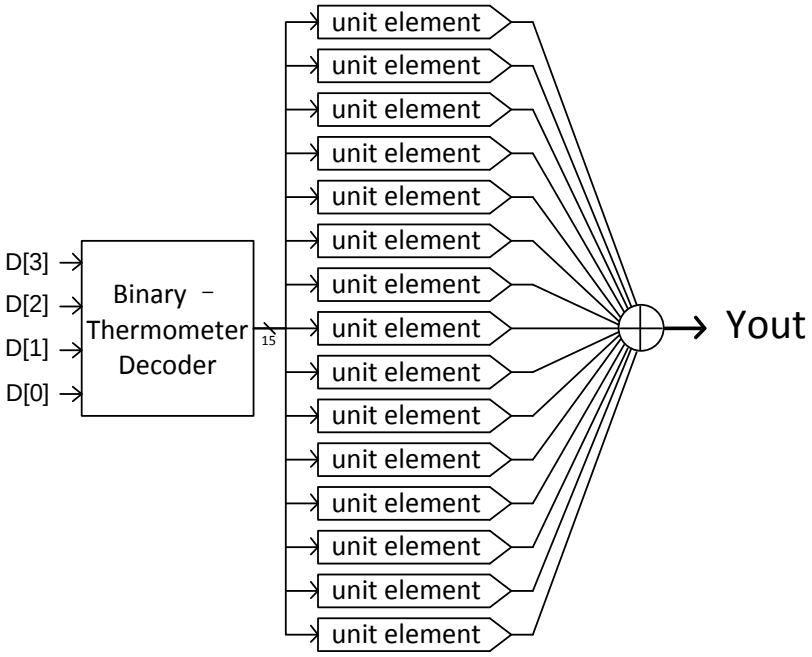


Fig. 1.2 4-bit thermometer DAC block diagram

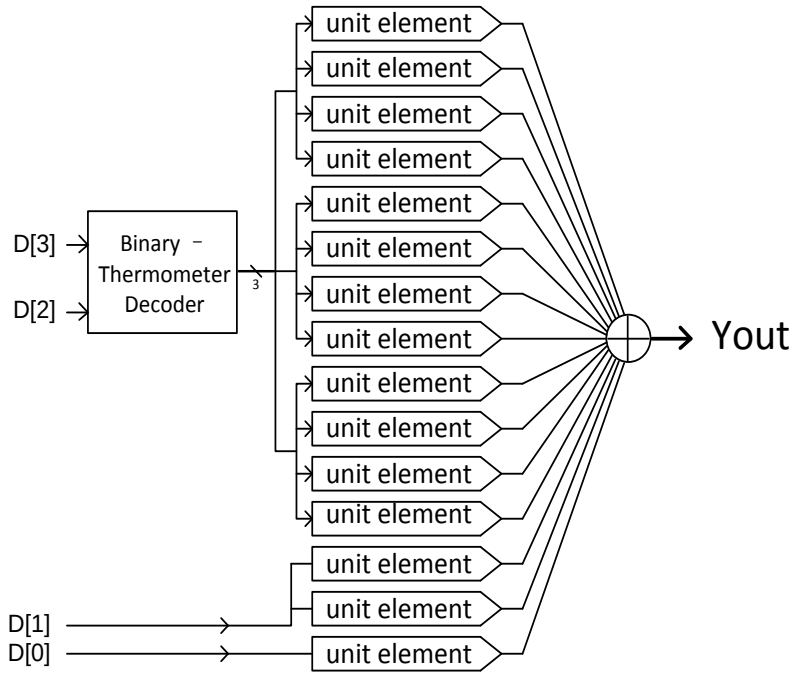


Fig. 1.3 4-bit segmented DAC block diagram

1.1 SOURCES OF NONLINEARITY IN DACS

Due to process variation, DACs inevitably suffer from the mismatch among the unit components. A basic way to relieve this issue is to group several unit element into a larger unit instead of using a single element in larger size (as shown in figures above). However, the effect of this method is not very satisfactory to achieve higher performance. To further address this problem, researchers have been working on various approaches to eliminate the mismatches and improve the linearity. Since mismatch error essentially results from the process variation, careful layout strategy is one of the straightforward methods to minimize the mismatches. Other than this, calibration and trimming are among the other popular ways to achieve higher linearity [2]. However, calibration and trimming are not so effective when dealing with signals in high frequency, because these two methods mainly reduce amplitude mismatches among elements. For high speed application, DEM becomes a popular approach to further reduce nonlinear distortion resulting from pulse shape and timing errors [3-11]. A detailed analysis of how DEM works is shown in [3-6]. The essential idea of DEM is to choose the elements in a pseudorandom pattern instead of thermometer-coded way. Thus, the distortions spread out and become uncorrelated with the input. Although the SNR will be reduced by using DEM, the total harmonic distortion (THD) can be greatly improved. Since linearity is what we care more in most cases of DACs, it is a worthwhile tradeoff. In Nyquist-rate high-resolution DACs, DEM can turn distortions resulting from mismatch error into white noise. The recent research published plenty of work demonstrating various DEM encoders to achieve higher signal-to-noise-and-distortion ratio (SNDR) or effective number of bits (ENOB) by mismatch scrambling.

1.2 INTRODUCTION TO NYQUIST-RATE DEM DACS

Fig. 1.1 illustrates that the binary-weighted DAC doesn't consist of a decoding block, which makes it simple to implement. However, component mismatches can bring in

severe linearity and even monotonicity (e.g. 01…11 to 10…00) issues in binary DACs. Also, its high switching activity leads to more power dissipation. [9] proposed the random rotation-based selection which alleviated nonlinearity and high switching activity and kept simplicity of binary-weighted DACs at the same time. The basic idea is shown as following:

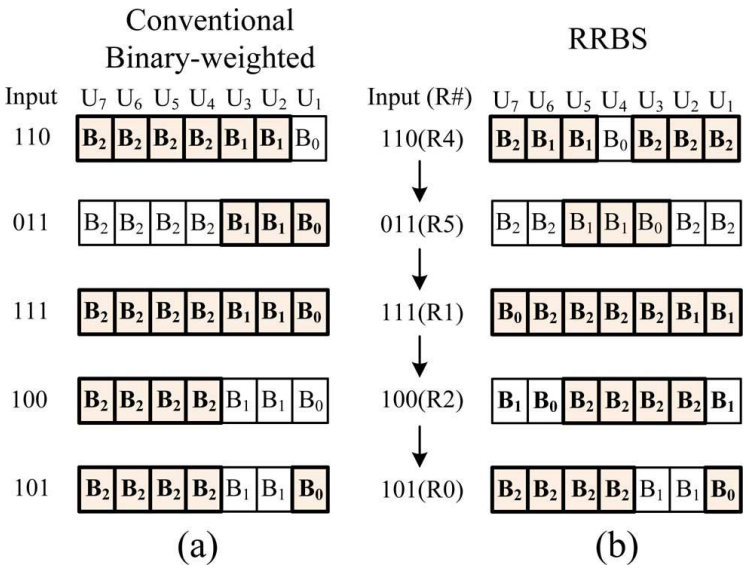


Fig. 1.4 Operating principal of the (a) conventional binary-weighted method and (b) random rotation-based binary-weighted selection (RRBS) method where R# represents the number of right-rotation steps [9]

Essentially, it shifts the usage of all elements from the conventional binary-weighted DAC. The number of steps the pattern is rotated is generated by a randomizer. Therefore, its DEM block has a low hardware cost. Similar idea can be implemented in a thermometer DAC [10]. Its operating principle is shown in below:

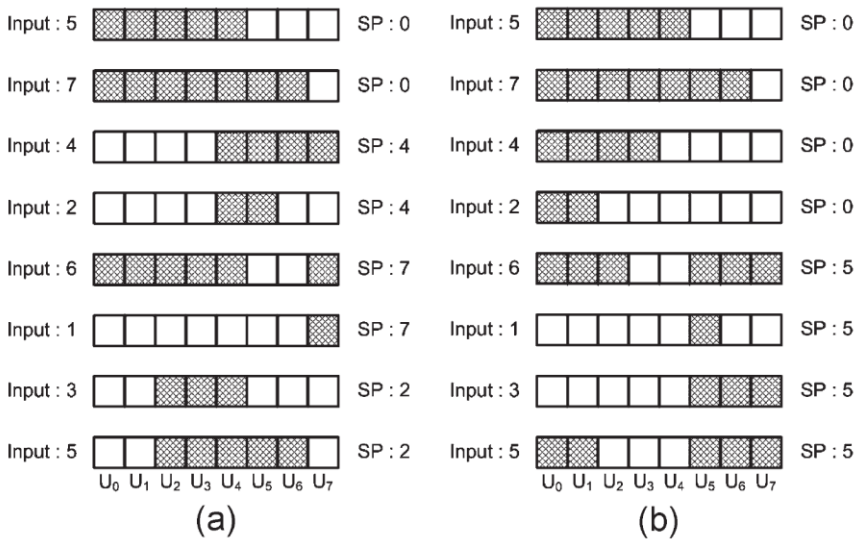


Fig. 1.5 Operation principles of the RTC method for (a) RP = 2 and (b) RP = 4 [10]

In Fig. 1.5, SP is the starting pointer pointing to the first element to choose. The randomization period (RP) indicates the frequency SP is changed. For example in the case in [9], RP is one, and SP changes for each sample. It is easy to understand that the smaller the RP is, more randomized the elements usage pattern will be, and more like white noise the mismatch spectrum will be. However, DACs implemented with a smaller RP will have higher switching activity and dissipate more power. The relationship between noise spectrum and RP can also be verified in simulation results shown below:

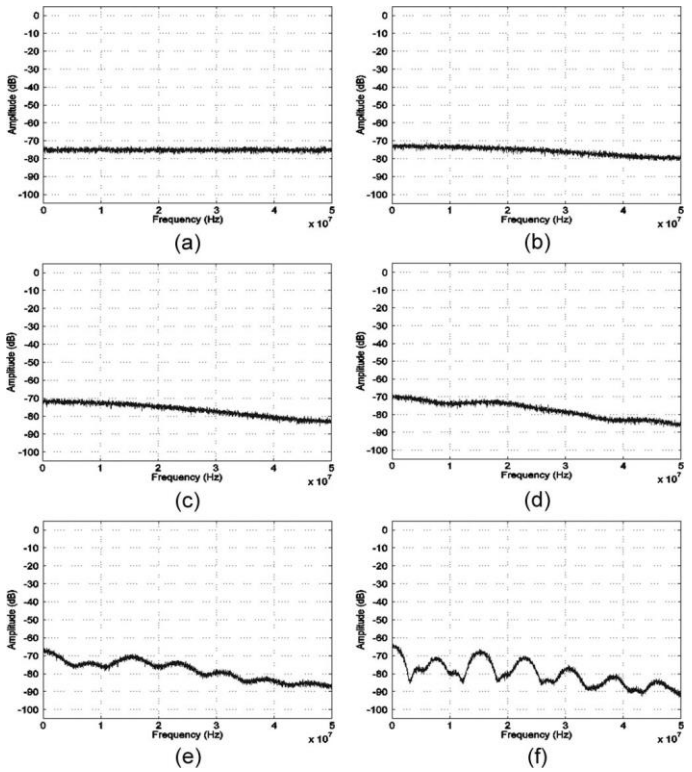


Fig. 1.6 Mismatch spectra of a DAC applying the RTC method with RP equal to (a) 1, (b) 2, (c) 4, (d) 8, (e) 16, and (f) 32 [10]

[11] proposed a high-resolution DEM DAC which achieves high linearity as well as high speed, at the expense of hardware complexity. The DEM DACs discussed above are summarized in the following table.

	[9]	[10]	[11]
Technology (μm)	0.18	0.18	0.18
Resolution (bits)	10	14	14
Sampling rate (MS/s)	500	10	100
SFDR (dB)	73	80.7	83
Area (mm^2)	0.034	0.28	3

Tab. 1.1 Comparison between DEM Nyquist-rate DACs

1.3 INTRODUCTION TO DELTA-SIGMA DACS

Nowadays, Delta-Sigma DACs become very popular in the case that the input signal has a limited bandwidth which is much lower than half of sampling frequency [12-13]. Thus, It is possible to take advantage of this property to shape the noise spectrum. The Delta-Sigma modulator (DSM) is the critical block to implement mismatch shaping that pushes most noise away from the frequency band we are interested in. The block diagram of a Delta-Sigma DAC is shown below.

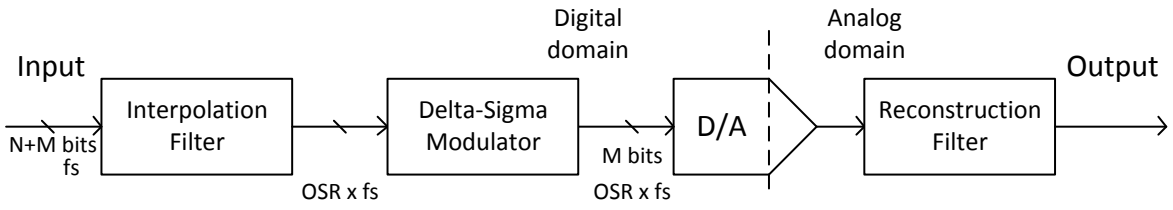


Fig. 1.7 Delta-Sigma DAC block diagram [13]

As shown above, the input is fed into an upsampling filter, typically an interpolator. After upsampling, the signal is at a higher frequency, and is fed into the Digital DSM. The modulator converts the high-resolution signal into a lower resolution. Meanwhile, the error from this resolution conversion is also taken back into the feedback loop which is inside the modulator. The last two stages are unit element DAC and reconstruction filter. All blocks except the last two stages are implemented in digital domain, which is favorable to accommodate semiconductor technology development. Furthermore, in analog domain, we only need a low-resolution but high-speed DAC, which does not consume high hardware complexity. The reconstruction filter, typically a lowpass filter, does not necessarily have very steep transition region and high stop band suppression. For DSMs, we define the order

of the loop filter as the order of the modulator itself. For example, a simple first-order delta-sigma modulator (DSM1) can be:

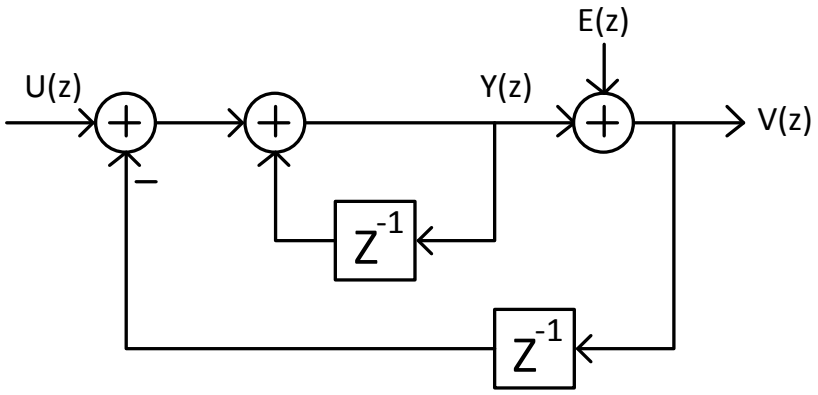


Fig. 1.8 Linear model of first-order Digital Delta-Sigma modulator

where E(z) is the quantization noise from bit resolution conversion.

From the model shown in Fig. 1.8, we can derive the following equations.

$$Y(z) = z^{-1}Y(z) + U(z) - z^{-1}V(z)$$

$$\begin{aligned} V(z) &= Y(z) + E(z) = z^{-1}Y(z) + U(z) - z^{-1}V(z) + E(z) \\ &= U(z) + E(z) - z^{-1}(V(z) - Y(z)) \\ &= U(z) + E(z) - z^{-1}E(z) \\ &= U(z) + (1 - z^{-1})E(z) \end{aligned}$$

We define signal transfer function (STF) and noise transfer function (NTF) as:

$$V(z) = STF(z)U(z) + NTF(z)E(z)$$

Then,

$$\begin{aligned} STF(z) &= 1 \\ NTF(z) &= 1 - z^{-1} \end{aligned}$$

Substitute z with $e^{j2\pi f}$, then the magnitude of NTF(z) will be:

$$\left|NTF(e^{j2\pi f})\right| = \left|2\sin(\pi f)\right|$$

where f is the frequency normalized by sampling frequency.

From equation above, we can see $|NTF|$ is increasing monotonically when normalized frequency f goes from 0 to 0.5. When $f << 1$, $|NTF| \approx 2\pi f$, based on Taylor expansion. The entire function curve is shown as following:

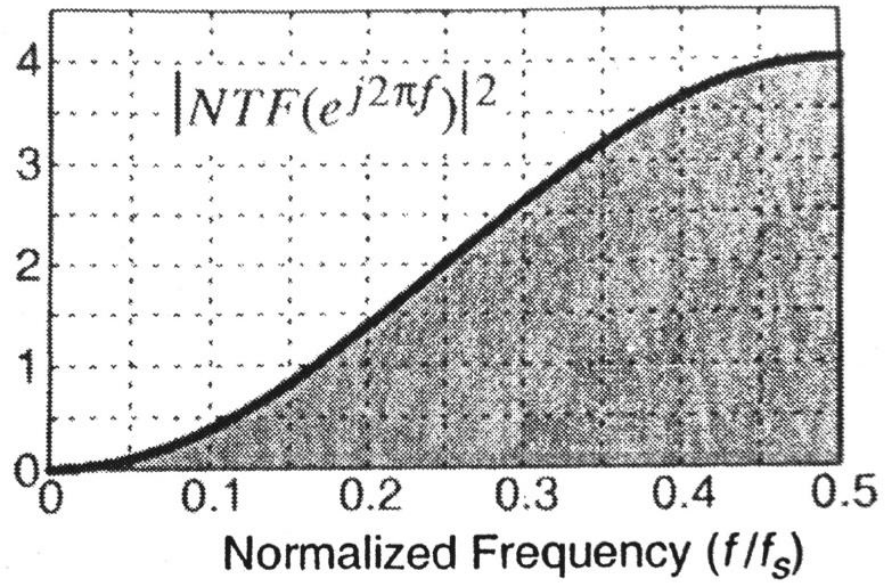


Fig. 1.9 Magnitude of NTF in DSM1 [13]

Given the advantages of DSMs, there are two ways to further increase the signal-to-noise-ratio (SNR). One is to raise the sampling frequency making the system operate at a higher rate and thus consumes more power. The other is to use a high-order NTF, but high-order DSMs often have stability concerns. An in-depth analysis is shown in [14] about the stability issue in DSMs and the relationship between modulator's order, OSR, and achievable SNR.

Although single-bit quantization is inherently linear, multibit quantization has lower quantization noise and better stability, leading to a higher SNR. Also, the output of

a multibit quantizer more closely resembles the desired analog signal in time domain, which releases the requirement on the reconstruction filter. Since components mismatch is an inevitable drawback leading to nonlinearity in multibit DSM, people have been working on algorithms to alleviate this impact.

In Chapter 2, many of the published algorithms on DEM in DAC are discussed. In Chapter 3, the structures which adopt mismatch shaping in non-segmented [24] and segmented [28] Delta-Sigma DAC are implemented logically and physically. The results from synthesis tools are analyzed in Chapter 3. Conclusion is presented in Chapter 4.

Chapter 2: DEM in Delta-Sigma DACs

2.1 BASIC STRUCTURE OF MISMATCH SHAPING DELTA-SIGMA DACS

To address the nonlinearity results from components mismatch, researchers introduced a block, Element Select Logic (ESL), to shape the mismatch noise spectrum in a similar way that DSMs shape the quantization noise. To achieve various features or/and alleviate different drawbacks, many algorithms on DEM in Delta-Sigma DACs have been proposed [15-31]. A theoretical analysis for multibit mismatch shaping DACs is done by J. Welz, and I. Galton [32]. A generic block diagram of a DEM Delta-Sigma DAC is shown as below:

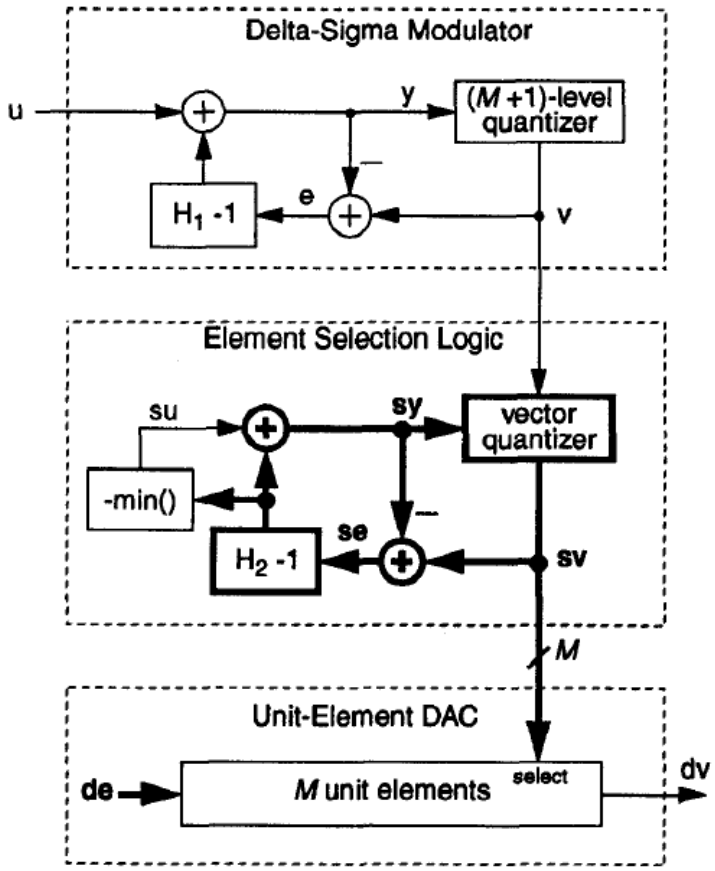


Fig. 2.1 Generic block diagram of a DEM Delta-Sigma DAC [16]

As shown above, ESL block takes the DSM's output, v[n], and generates a vector, $\overrightarrow{sv}[n]$, which consists of M bits. Each bit of $\overrightarrow{sv}$ drives one element in the DAC. The conversion from v[n] to $\overrightarrow{sv}[n]$ should meet the following equation:

$$\sum \overrightarrow{sv}[n] = v[n], \qquad \forall n.$$

DWA is a popular algorithm to implement this first-order mismatch shaping feature [17-21]. Zhang and Temes proposed one type of DWA, called SeDWA [18], which divided the DAC elements into several sets and conduct the rotation-based element usage pattern within each set. As expected, components are used more randomly and inband tones are reduced. Thus, Higher SFDR can be achieved.

2.2 STABILITY ISSUES AND SOLUTIONS IN ESL

As we can see from the Fig. 2.1, $\overrightarrow{se}$ subtracts its minimum value before getting fed back to the vector quantizer. This is for stability concerns and used to hold the minimum value of the feedback vector zero [22-23]. However, it costs high hardware complexity to search a vector's minimum value. A simpler structure that can hold the vector's average constant is proposed in [24].

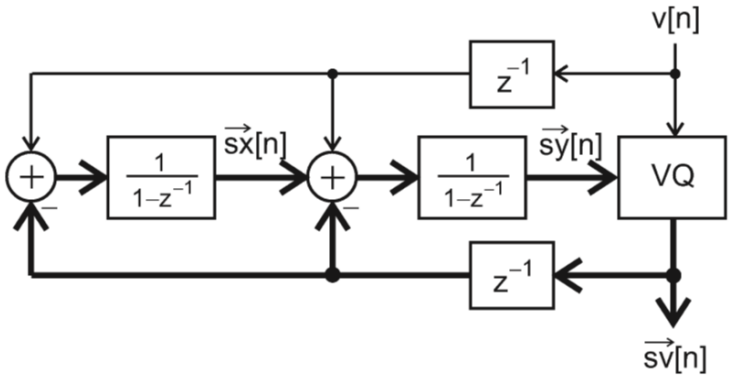


Fig. 2.2 Modified ESL architecture that holds the vector average constant [24]

Although the structure in Fig. 2.2 is easier to implement, it can still go unbounded when the input has a large amplitude. In [24], a new vector quantizer is proposed that takes the linear combination of both $\overrightarrow{sy}[n]$ and $\overrightarrow{sx}[n]$, instead of only $\overrightarrow{sy}[n]$ in Fig. 2.2.

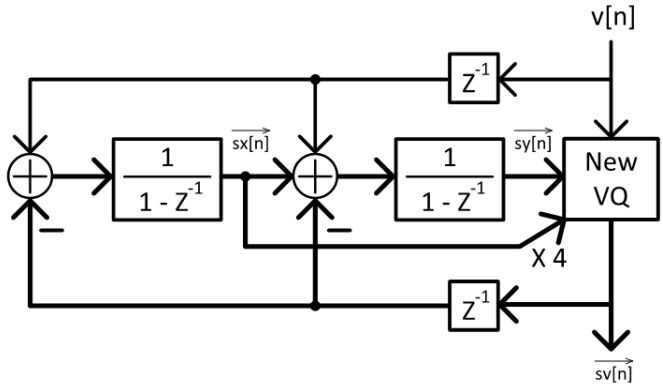


Fig. 2.3 The stable second-order mismatch shaping ESL structure

As shown in Fig. 2.3, this new VQ takes $\overrightarrow{sy}[n] + 4\overrightarrow{sx}[n]$ as its input. A comparison between these two architectures is done through simulation with the input that has an amplitude A of 0.85. The result is shown as below [24]:

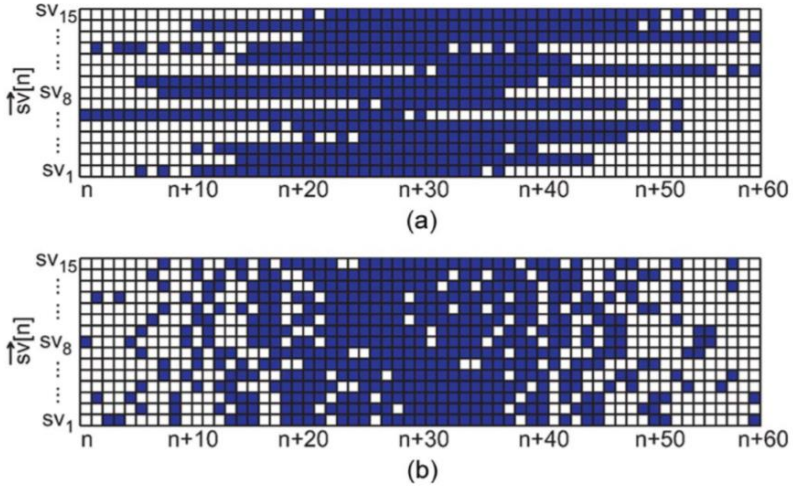


Fig. 2.4 Data usage pattern with: (a) conventional VQ; (b) new VQ [24]

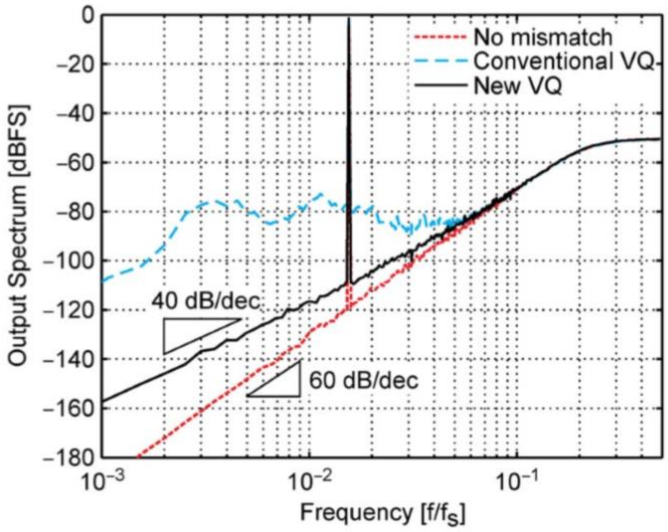


Fig. 2.5 Output spectrum for the case of Fig. 2.4 [24]

Fig. 2.2 shows a thermometer-like element usage pattern and its output spectrum doesn't have any mismatch shaping in the lower band, which indicates that it goes unstable for large input. Meanwhile, the new VQ system has a more random element usage sequence and it shows second-order (40dB/dec) mismatch shaping at the low frequencies in its output spectra.

By selecting the proper coefficients for the linear combination of the state vectors, the structure shown in Fig. 2.3 can be expanded to achieve higher-order mismatch shaping ESL. The following combination of the state vectors is verified in [24] to be stable in the third-order and forth-order mismatch shaping ESL.

$$\text{Third-order: } \vec{R}[n] = \overrightarrow{sz}[n] + 16\overrightarrow{sy}[n] + 64\overrightarrow{sx}[n]$$

$$\text{Fourth-order: } \vec{R}[n] = \overrightarrow{sw}[n] + 16\overrightarrow{sz}[n] + 64\overrightarrow{sy}[n] + 512\overrightarrow{sx}[n]$$

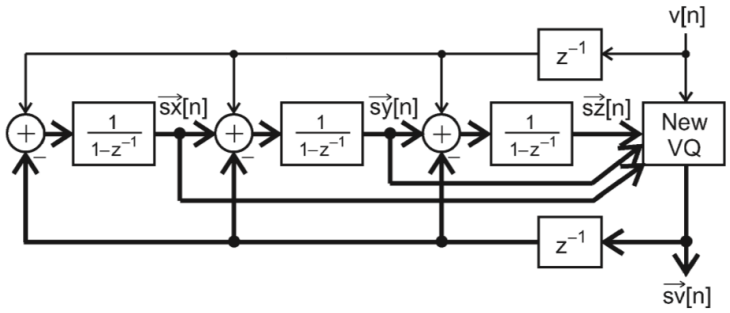


Fig. 2.6 Stable third-order mismatch-shaping ESL in [24]

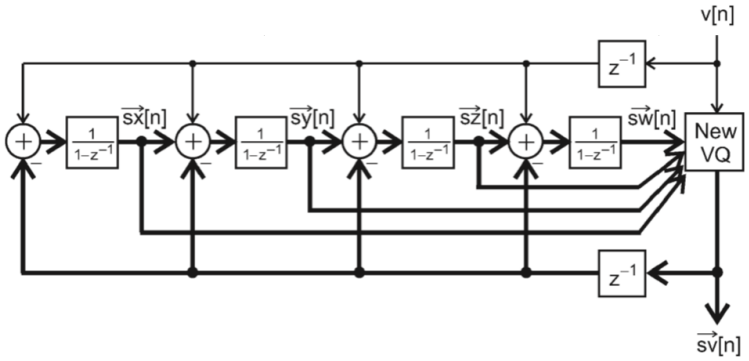


Fig. 2.7 Stable forth-order mismatch-shaping ESL in [24]

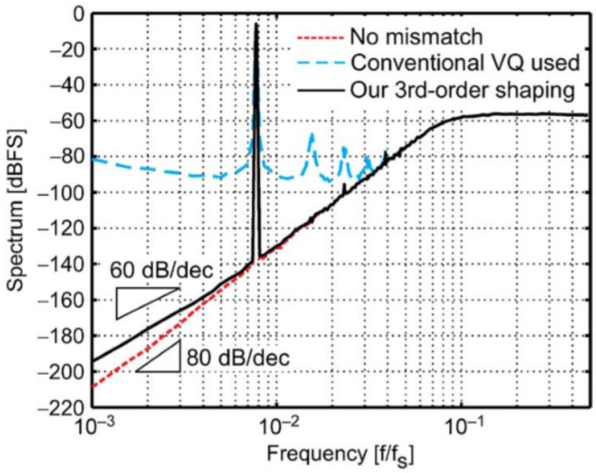


Fig. 2.8 Comparison between third-order mismatch-shaping VQ in [24] and conventional VQ for A = 0.5

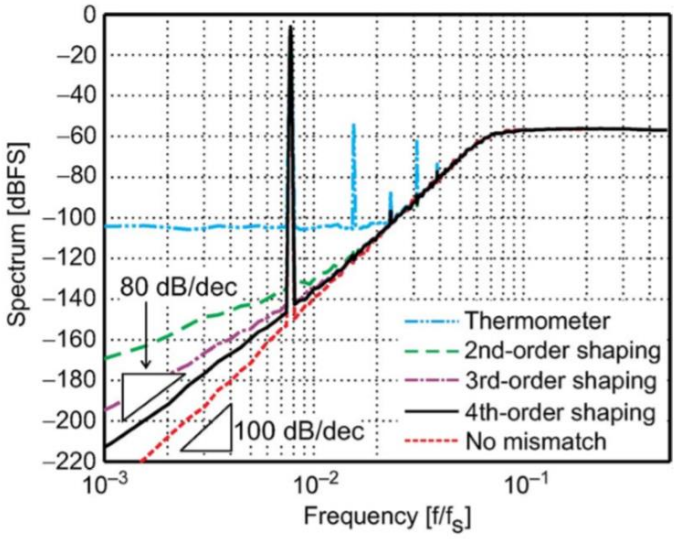


Fig. 2.9 Comparison among different ESL in [24]

Both Fig. 2.8 and Fig. 2.9 indicate that the third-order and forth-order ESL can keep stable with the new VQ taking a linear combination of all intermediate state vectors, while the conventional VQ goes unbounded. Moreover, all coefficients adopted above can be easily implemented by bit shifting since they are all power of 2.

2.3 SEGMENTED MISMATCH SHAPING ESL ALGORITHM

Other than the stability issue, we can see that the complexity of ESL and the number of unit elements grow exponentially with DAC's bit width. Therefore, this DAC usually is less than 5 bits, given the reasonable area and power budget. To further increase the bit-width of the DAC, it is a straightforward idea to make it a segmented DAC like we did for a traditional Nyquist-rate DAC. In other words, a segmented DAC can reduce the number of unit elements and control signals need to be generated by the ESL. For example, if we want to design a 6-bit DAC, a segmented one can have three segments, with a weight of 1, 4, and 16, respectively. Each segment has 4 units. In total, ESL needs 3 x 4 = 12 output control signals, comparing to 63 signals for its counterpart with all unit-weight elements.

To incorporate mismatch shaping into a segmented DAC, several ESLs are needed to control separate segments and shape the mismatch error within each segment. However, element mismatch between any two of the segments are not alleviated by simply breaking ESL into several parts. To deal with mismatch between different segments, [25-26] proposed the tree-structured segmented the mismatch shaping DAC and analyzed systematically the techniques to design mismatch shaping Delta-Sigma DACs. In this structure, the elements in all segments is required to be weighting the power of two. Its mismatch shaping order cannot be higher than two.

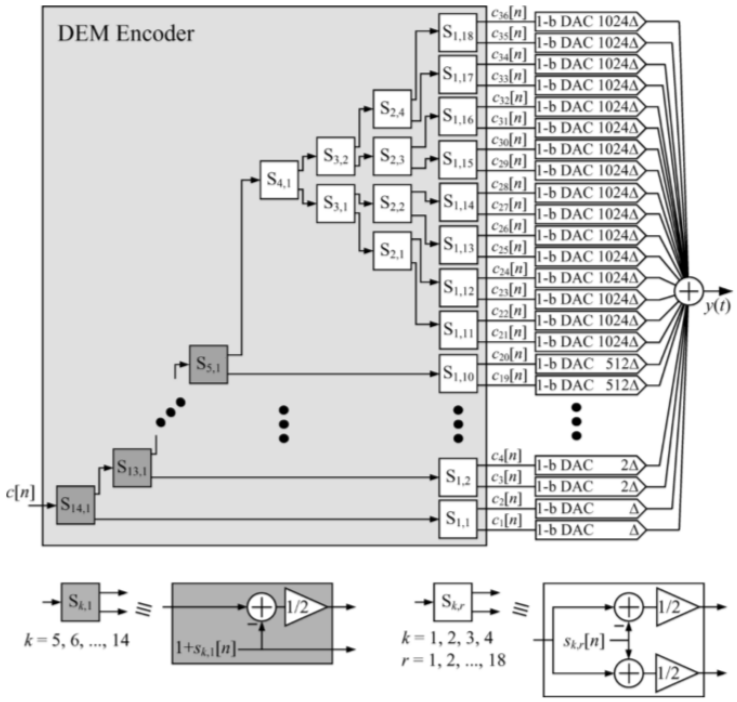


Fig. 2.10 The architecture of segmented DAC proposed in [25]

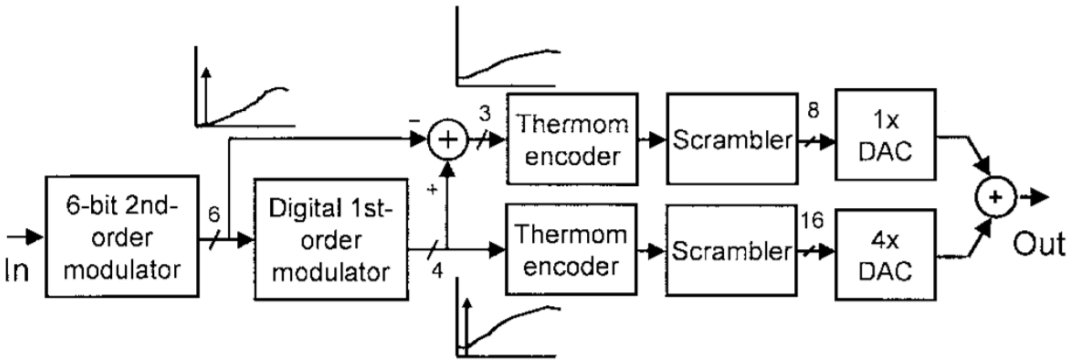


Fig. 2.11 Segmented mismatch shaping DAC in [27]

[27] proposed an approach to achieve a higher-order segmented mismatch shaping DAC (Fig. 2.11), and it doesn't set strict requirement on the unit element weights. However, its higher-order mismatch shaping is obtained at the expense of a larger number of unit elements.

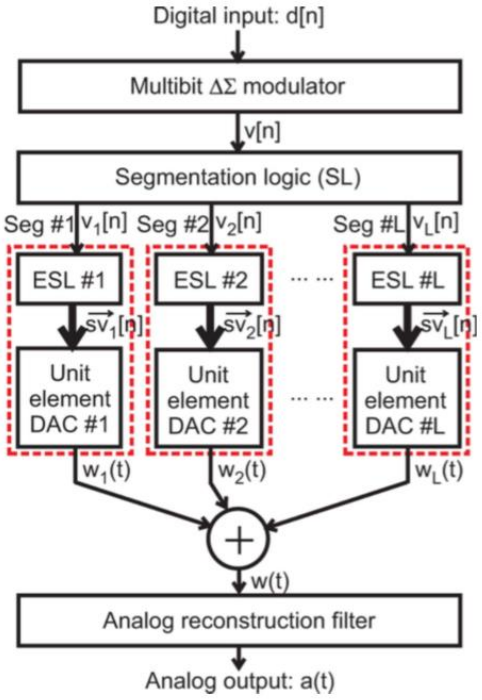
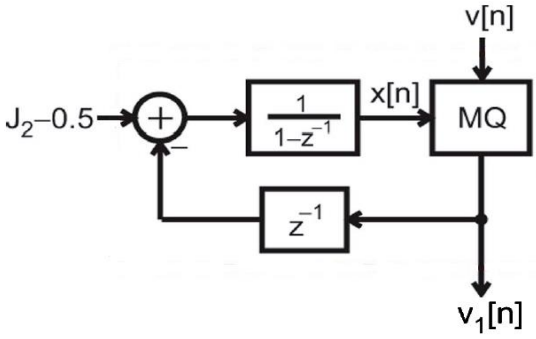
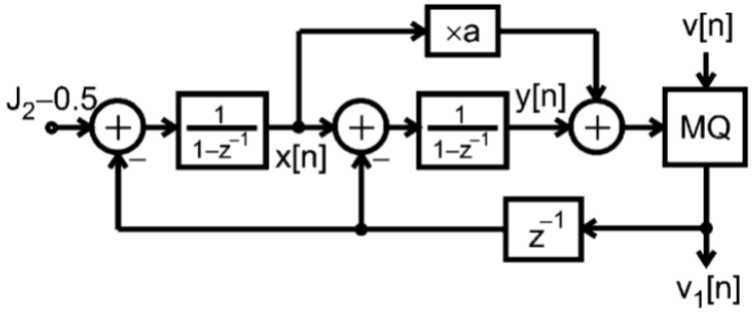


Fig. 2.12 Block diagram of segmented mismatch shaping DAC in [28]

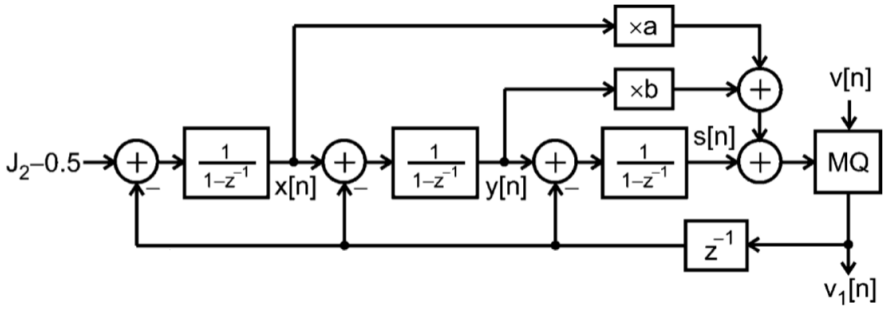
Fig. 2.12 shows the architecture of the segmented mismatch shaping DAC published in [28], which overcomes the down sides of those in [25-27]. The segmentation logic (SL) block is incorporated in this block to deal with mismatch among segments. The way to generate $v_1[n]$, $v_2[n]$, …, $v_L[n]$ feeding into each of segments is shown in below:



(a)



(b)



(c)

Fig. 2.13 Generation of $v_1[n]$ with (a) first-order mismatch shaping (b) second-order mismatch shaping (c) third-order mismatch shaping [28]

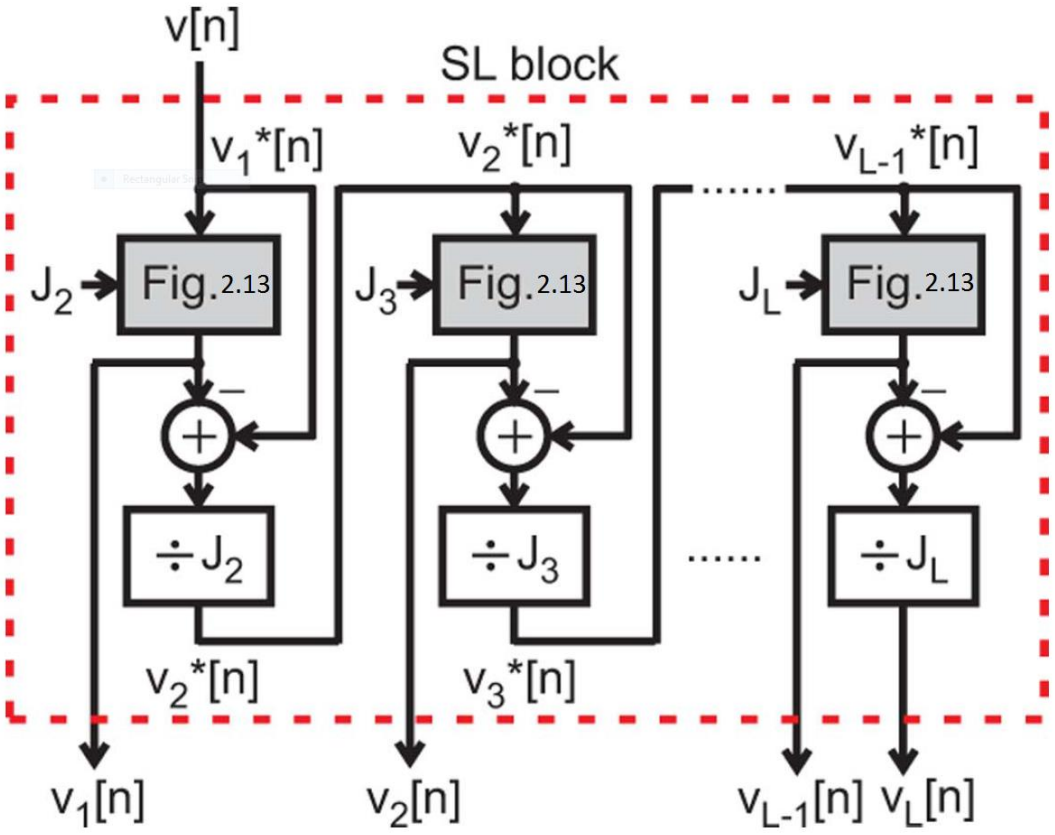


Fig. 2.14 Structure of a complete SL assembled iteratively [28]

Fig. 2.13 and Fig. 2.14 show the structure of SL with first-order, second-order, and third-order mismatch shaping. Taking the stability, noise surrpression, and hardware cost into consideration, we choose a = 1 for the second-oder SL and a = 32, b = 8 for the third-order version (See Fig. 2.13 (b) and (c)). The output of modular quantizer (MQ), $v_1[n]$, is dependent on the polarity of its input, R[n] for instance. This dependence is defined as:

$$v_1[n] = \begin{cases} v[n] \bmod J_2, & R[n] < 0 \\ (v[n] \bmod J_2) + J_2, & R[n] \geq 0 \end{cases}$$

It is analyzed theoretically and verified through simulation that the number of unit components does not increase as the order of mismatch shaping goes up [28].

Chapter 3: DEM Implementation and Results

In this chapter, the algorithms introduced in [24] and [28] are implemented in Verilog and synthesized through Design Compiler. With the interconnection taken into account, physical layout is also complemented to give us more accurate timing, power and area information.

3.1 LOGIC IMPLEMENTATION OF NON-SEGMENTED ESL:

The Verilog implementation for ESL block is shown below:

```
begin
    sx[0] <= sx[0] - sv[0] + v;
    sx[1] <= sx[1] - sv[1] + v;
    sx[2] <= sx[2] - sv[2] + v;
    sx[3] <= sx[3] - sv[3] + v;
    sx[4] <= sx[4] - sv[4] + v;
    sx[5] <= sx[5] - sv[5] + v;
    sx[6] <= sx[6] - sv[6] + v;
    sy[0] <= sy[0] + sx[0] - sv[0] + v;
    sy[1] <= sy[1] + sx[1] - sv[1] + v;
    sy[2] <= sy[2] + sx[2] - sv[2] + v;
    sy[3] <= sy[3] + sx[3] - sv[3] + v;
    sy[4] <= sy[4] + sx[4] - sv[4] + v;
    sy[5] <= sy[5] + sx[5] - sv[5] + v;
    sy[6] <= sy[6] + sx[6] - sv[6] + v;
    sz[0] <= sz[0] + sy[0] - sv[0] + v;
    sz[1] <= sz[1] + sy[1] - sv[1] + v;
    sz[2] <= sz[2] + sy[2] - sv[2] + v;
    sz[3] <= sz[3] + sy[3] - sv[3] + v;
    sz[4] <= sz[4] + sy[4] - sv[4] + v;
    sz[5] <= sz[5] + sy[5] - sv[5] + v;
    sz[6] <= sz[6] + sy[6] - sv[6] + v;
    sw[0] <= sw[0] + sz[0] - sv[0] + v;
    sw[1] <= sw[1] + sz[1] - sv[1] + v;
    sw[2] <= sw[2] + sz[2] - sv[2] + v;
    sw[3] <= sw[3] + sz[3] - sv[3] + v;
    sw[4] <= sw[4] + sz[4] - sv[4] + v;
    sw[5] <= sw[5] + sz[5] - sv[5] + v;
    sw[6] <= sw[6] + sz[6] - sv[6] + v;
end
```

Added for 3rd-order

Added for 4th-order

Fig. 3.1 Verilog implementation of mismatch shaping ESL

```
assign R[0] = sy[0] + (sx[0] << 2);
assign R[1] = sy[1] + (sx[1] << 2);
assign R[2] = sy[2] + (sx[2] << 2);
assign R[3] = sy[3] + (sx[3] << 2);
assign R[4] = sy[4] + (sx[4] << 2);
assign R[5] = sy[5] + (sx[5] << 2);
assign R[6] = sy[6] + (sx[6] << 2);
```

(a)

```
assign R[0] = sz[0] + (sy[0] << 4) + (sx[0] << 6);
assign R[1] = sz[1] + (sy[1] << 4) + (sx[1] << 6);
assign R[2] = sz[2] + (sy[2] << 4) + (sx[2] << 6);
assign R[3] = sz[3] + (sy[3] << 4) + (sx[3] << 6);
assign R[4] = sz[4] + (sy[4] << 4) + (sx[4] << 6);
assign R[5] = sz[5] + (sy[5] << 4) + (sx[5] << 6);
assign R[6] = sz[6] + (sy[6] << 4) + (sx[6] << 6);
```

(b)

```
assign R[0] = sw[0] + (sz[0] << 4) + (sy[0] << 6) + (sx[0] << 9);
assign R[1] = sw[1] + (sz[1] << 4) + (sy[1] << 6) + (sx[1] << 9);
assign R[2] = sw[2] + (sz[2] << 4) + (sy[2] << 6) + (sx[2] << 9);
assign R[3] = sw[3] + (sz[3] << 4) + (sy[3] << 6) + (sx[3] << 9);
assign R[4] = sw[4] + (sz[4] << 4) + (sy[4] << 6) + (sx[4] << 9);
assign R[5] = sw[5] + (sz[5] << 4) + (sy[5] << 6) + (sx[5] << 9);
assign R[6] = sw[6] + (sz[6] << 4) + (sy[6] << 6) + (sx[6] << 9);
```

(c)

Fig. 3.2 Linear combination of state vectors for ESL with (a) 1st order (b) 2nd order (c) 3rd order

As described in Chapter 2, the linear combination, $\vec{R}$ in Fig. 3.2, of these state vectors, which are, $\overrightarrow{sx}$, $\overrightarrow{sy}$, $\overrightarrow{sz}$ and $\overrightarrow{sw}$, is fed into another module, VQ (see Fig. 2.3). To simplify its calculation, we choose all coefficients to be the power of 2, so that multiplication can be converted into shifting. The module VQ basically does comparison among all the values of $\vec{R}$. With the sequence information of state vectors, VQ finds the top "v" values in $\vec{R}$ and sets the according elements in $\overrightarrow{sv}$ to be "1". For example, the vector $\vec{R}$ has M datas in total, which are {R[0], R[1], …, R[M-1]}. In a certain cycle, the output of the DSM, v, is 4. In this cycle, R[1], R[4], R[6], and R[M-1] are the four largest values. Therefore, sv[1], sv[4], sv[6], and sv[M-1] will be set to be "1" for this cycle. In other

word, the essential function in VQ is a partial sorter, which is targeted to find the indexes of the top n values in the vector. This mechanism in VQ automatically makes our ESL meets the following requirement.

$$\sum \overrightarrow{sv}[n] = v[n], \qquad \forall n$$

3.2 LOGIC SYNTHESIS AND PHYSICAL IMPLEMENT OF NON-SEGMENTED ESL:

With standard cell library, synthesized netlists of ESL in different order are generated by Synopsys' Design Compiler. Based on that, the physical layout is completed using IC Compiler with TSMC 0.18 μm process. In this part, we implemented a non-segmented ESL for DACs with seven elements. The figure below shows the final layout of the second-order ESL, which has a total area of 270 μm x 270 μm.

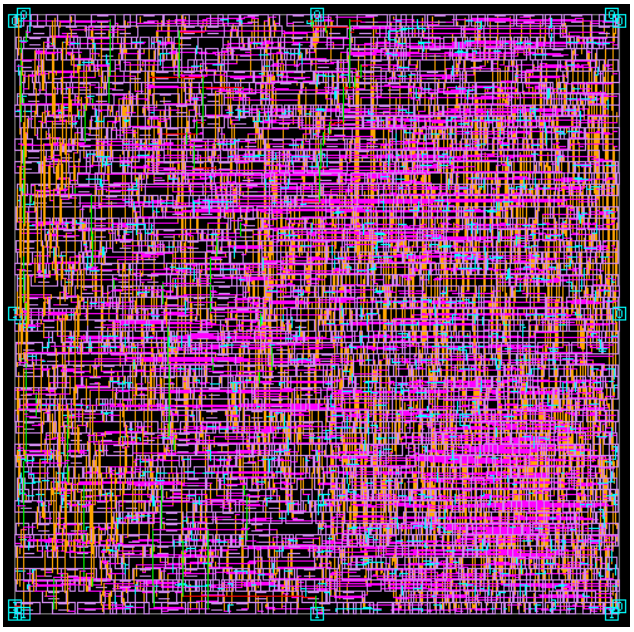


Fig. 3.3 Layout of the second-order ESL (power grid not shown)

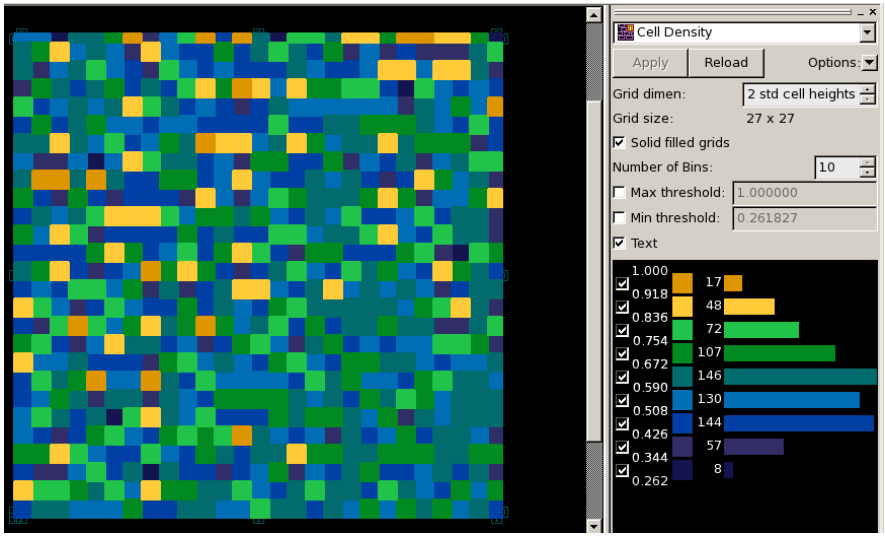


Fig. 3.4 Cell density of the second-order ESL

Fig. 3.4 shows the cell density on the placement of 2nd-order ESL. We can see the average utilization is around 60%, which is within the reasonable range. The following figure shows its critical path, which has a positive timing slack indicating we meet the target frequency, 50MHz.

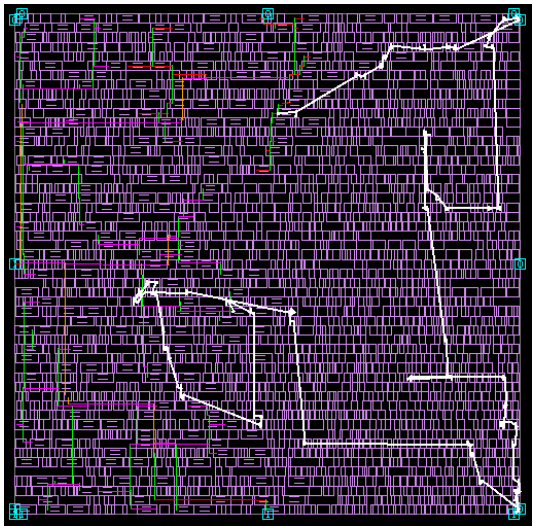


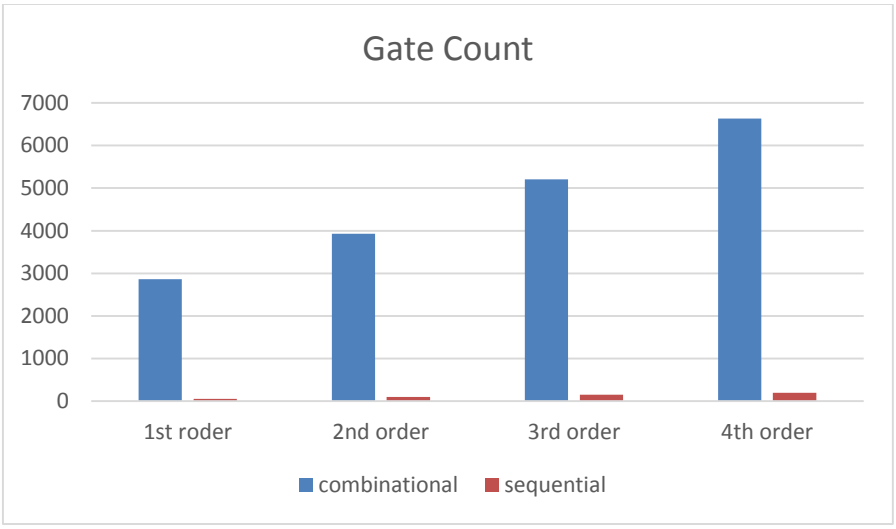
Fig. 3.5 Critical path of the second-order ESL

Different mismatch shaping order ESLs operating at 50MHz are compared as below in terms of area (gate count), power, and:

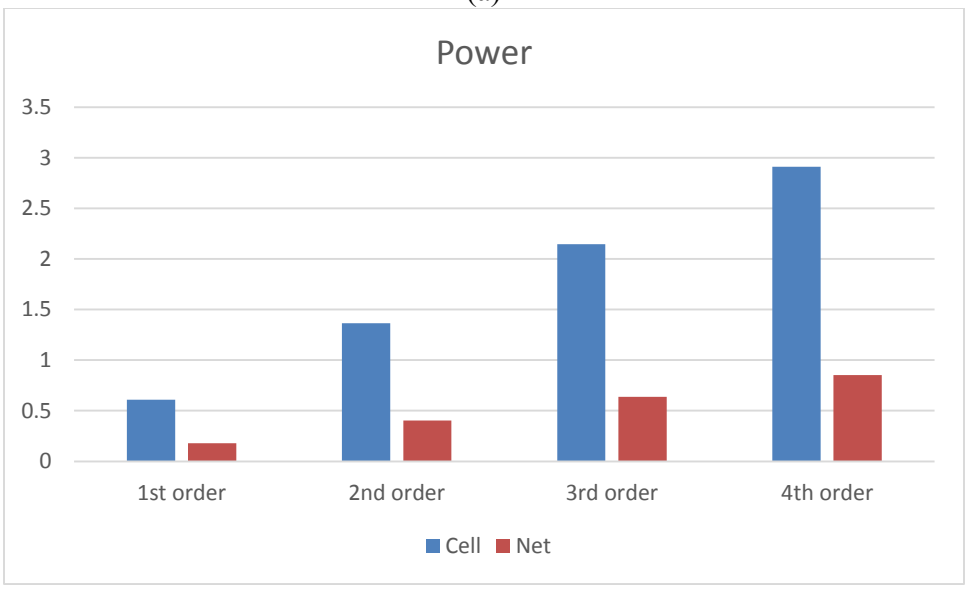
	Gate Count		Total Cell Area (μm^2)	Power (mW)			Delay (ns)
	Combinational	Sequential		Cell	Net	Total	
1st Order	2859	52	52984	0.61	0.18	0.79	8.04
2nd Order	3928	98	70303	1.36	0.40	1.76	8.05
3rd Order	5205	147	98933	2.14	0.64	2.78	8.02
4th Order	6635	196	128638	2.91	0.85	3.76	8.06

Tab. 3.1 Comparison between ESLs with different order

Tab. 3.1 shows ESLs in different orders have similar delay. This can be explained by their critical path, which starts from the flop that stores $\vec{R}$, goes through VQ, and eventually ends up to the output pin (see Fig. 3.5). VQ in these three ESLs are almost the same, so it introduces same amount of delay in ESL.



(a)



(b)

Fig. 3.6 Breakdown figure of ESL on (a) gate count (b) power

Fig. 3.6 gives a more straight forward view about the percentage of different components in delay and power.

3.3 LOGIC IMPLEMENTATION OF SEGMENTED ESL:

The segmented mismatch shaping algorithm in [28] is also implemented in the same way and synthesis environment. As shown in the block diagram of the segmented ESL (Fig. 2.12), the implementation of ESL described above can be reused to assemble this segmented version. The critical part is within the SL block, and its code is shown in the following:

```
always @ (posedge clk)
begin
  if(!rst)
    x <= 0;
  else
    x <= x - r_v1 + 3'b111;  //J - 0.5 = 3.5
end

always @ (posedge clk)
begin
  if(!rst)
    y <= 0;
  else
    y <= y + x - r_v1;
end

always @ (posedge clk)
begin
  if(!rst)
    s <= 0;
  else
    s <= s + y - r_v1;
end
```

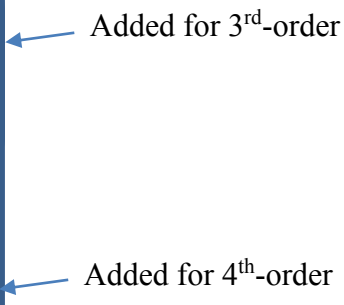


Fig. 3.7 Implementation of mismatch shaping SL

For SL in different order, the MQ module (see Fig. 2.13) keeps the same. Its main idea is to get the residue and add J to it when x is larger or equal to zero. In this paper, we take J equals to 4.

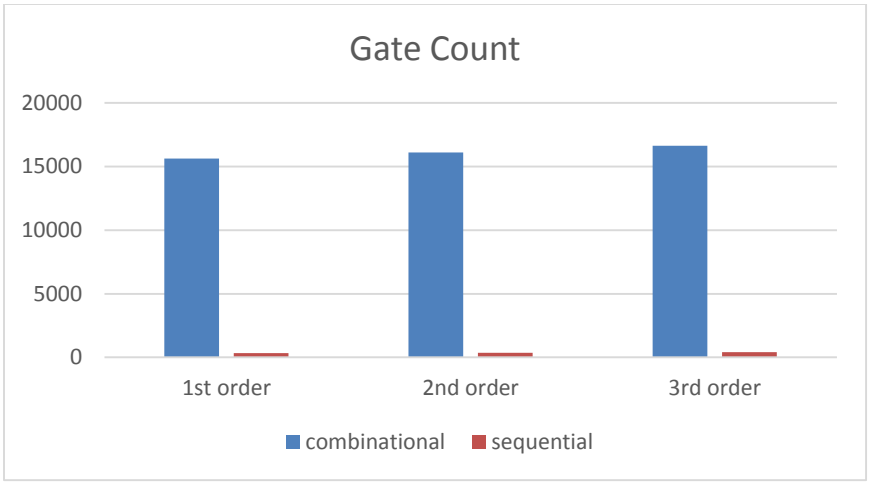
3.4 LOGIC SYNTHESIS AND PHYSICAL IMPLEMENT OF SEGMENTED ESL:

The physical layout for segmented ESL can be built up based on its block diagram. So these submodules are clustered in blockages when doing placement. With its physical information, we did the similar comparison among segmented ESL blocks in different order as the comparison between ESL blocks.

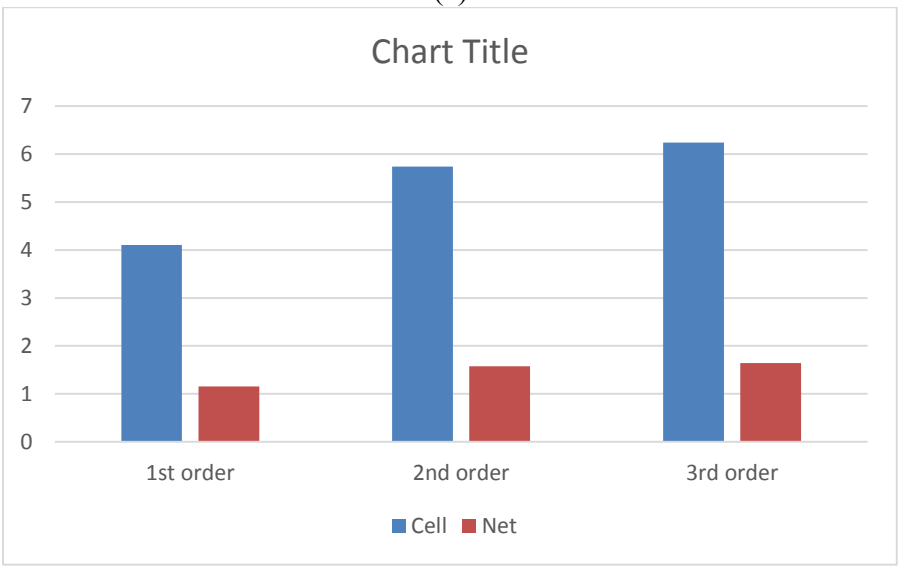
	Gate Count		Total Cell Area (μm^2)	Power (mW)			Delay (ns)
	Combinational	Sequential		Cell	Net	Total	
1st Order	15617	336	280621	4.11	1.16	5.26	9.04
2nd Order	16111	354	295327	5.74	1.58	7.32	9.20
3rd Order	16638	400	300613	6.24	1.64	7.88	9.16

Tab. 3.2 Comparison between segmented ESL with different order

As we explained in the previous part, the speed of segmented ESLs doesn't vary a lot among different mismatch shaping orders. Moreover, we can see it from Tab. 3.2 that the combinational cell number increases by only 6.5% when mismatch shaping order grows up to the third order, while the sequential cell number increases by almost 20%. This difference is mainly due to two reasons: the partial sorter in VQ is the dominant part in gate count of the whole ESL; the number of elements that need to be sorted doesn't change with the order of ESL. In other words, VQ can be reused in ESLs with different mismatch shaping orders. Therefore, the bit-width of ESL is the dominants its complexity.



(a)



(b)

Fig. 3.8 Breakdown figure of ESL on (a) gate count (b) power

As we described in Chapter 2, the segmented mismatch shaping block has an advantage in hardware complexity, compared to its non-segmented counterpart with the same number of quantization levels. It's predictable that the complexity of a non-segmented increases exponentially with increasing in bits of quantization levels.

$Total\ Gate\ Count \propto 2^B$, where B is the resolution of DACs

If we implement a 7-bit non-segmented ESL in the same way as what we did in section 3.1 and 3.2, its hardware cost can be derived based on the relation shown above. The comparison between these two algorithms is shown as below:

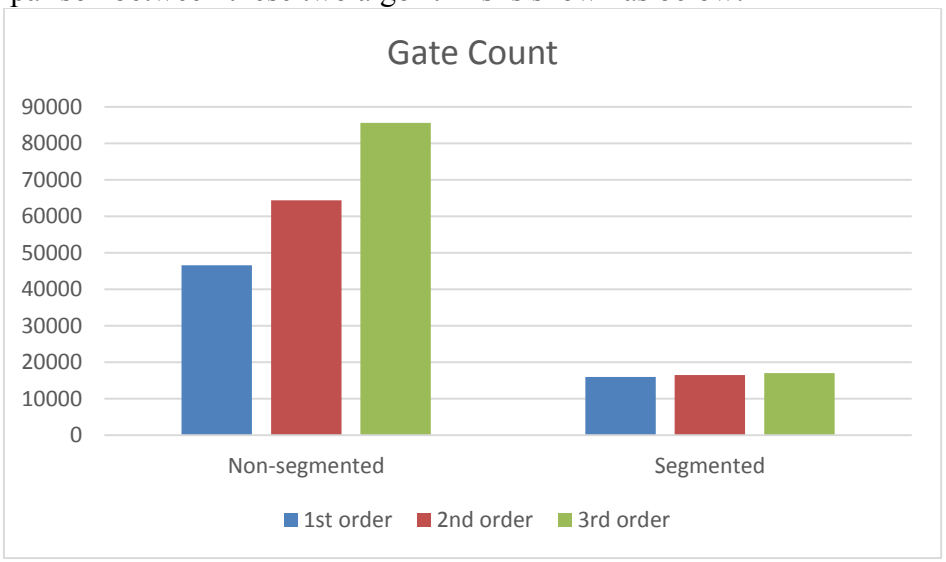


Fig. 3.9 Comparison on gate count between segmented and non-segmented ESL with same levels of quantization

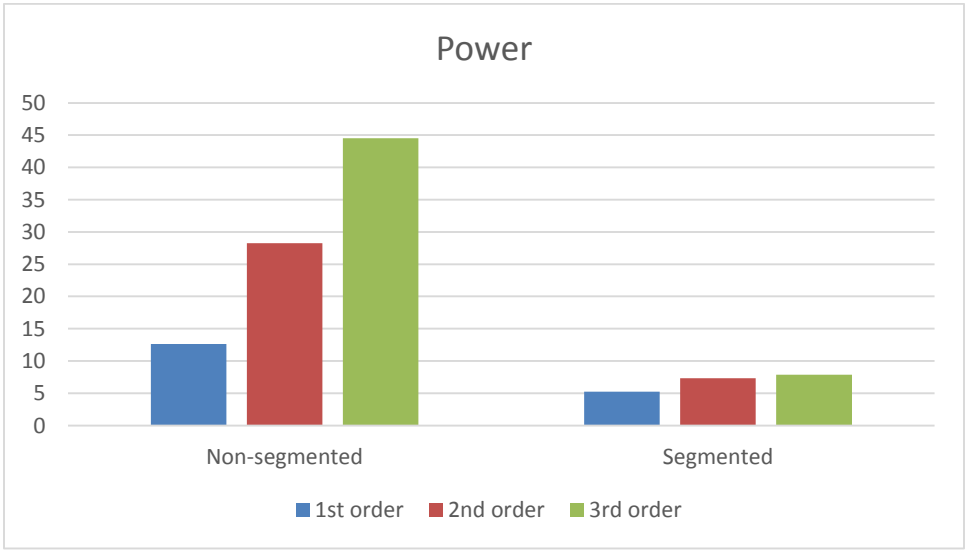


Fig. 3.10 Comparison on power between segmented and non-segmented ESL with same levels of quantization

Chapter 4: Conclusion

From the results in Chapter 3, higher-order mismatch shaping in ESL can give better noise suppression but needs more hardware cost. One obvious reason is that it requires more sequential cells to restore intermediate state vectors. Moreover, the linear combination of the state vectors requires more and wider adders as the mismatch shaping order raises. Therefore, there is a tradeoff between mismatch shaping order and hardware complexity. Designing mismatch shaping DACs can be targeted with specification on various aspects. A mismatch noise suppression driven design usually consumes more area and dissipates more power.

Instead of increasing the cell number exponentially to achieve more quantization levels, segmented ESL alleviates the complexity issue. In this thesis, we implemented a segmented ESL which can achieve higher order mismatch shaping and hold the number of unit elements in DAC as mismatch shaping order increases.

References

[1] Rudy van de Plassche, “CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters, 2nd Edition” Chapter 4, Kluwer Academic Publishers, 2003

[2] Franco Maloberti, “Data Converters” Chapter 3, Published by Springer, 2007

[3] Ian Galton, “Why dynamic-element-matching DACs work," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 57, pp. 69-74, Feb. 2010

[4] P. Rombouts and L. Weyten, “A study of dynamic element-matching techniques for 3-level unit elements,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 47, no. 11, pp. 1177-1187, Nov. 2000

[5] Mohammad Taherzadeh-Sani and Anas A. Hamoui, “Analysis of dynamic element matching (DEM) in pipelined ADCs,” Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 5263 - 5266, May 2006

[6] I. Galton and P. Carbone, “A rigorous error analysis of D/A conversion with dynamic element mismatching,” IEEE Transactions on Circuits and Systems II: Analog and Digital Processing, vol. 42, no. 12, pp. 763-772, Dec. 1995

[7] Nan Sun, “Low-complexity high-order vector-based mismatch shaping in multibit $\Delta\sum$ ADCs,” IEEE IEEE Transactions Circuits and System II: Express Briefs, vol. 58, no. 12, pp. 872-876, Dec. 2011

[8] Akira Yasuda, Hiroshi Tanimoto, and Tetsuya Iida, “A third-order Delta-Sigma modulator using second-order noise-shaping dynamic element mismatching,” IEEE Journal of Solid-State Circuits, vol. 33, no. 12, pp. 1879-1886, Dec. 1998

[9] Wei-Te Lin and Tai-Haur Kuo, “A compact dynamic-performance-improved current-steering DAC with random rotation-based binary-weighted selection,” IEEE Journal of Solid-State Circuits, vol. 47, no. 2, pp. 444-453, Feb. 2012

[10] Da-Huei Lee, Tai-Haur Kuo, and Kow-Liang Wen, “Low-cost 14-bit current-steering DAC with a randomized thermometer-coding method,” IEEE Transactions Circuits and System II: Express Briefs, vol. 56, no. 2, pp. 137-141, Feb. 2009

[11] Kok Lim Chan , Jianyu Zhu and Ian Galton "Dynamic element matching to prevent nonlinear distortion from pulse-shape mismatches in high-resolution DACs", IEEE J. Solid-State Circuits, vol. 43, no. 9, pp.2607 -2078, Sep. 2008

[12] Richard Gaggl, “Delta-Sigma A/D Converters,” Published by Springer, 2013

[13] Richard Schreier and Gabor C. Temes, “Understanding Delta-Sigma Data Converters,” Wiley-IEEE Press, November 2004

[14] Richard Schreier, “An empirical study of high-order single-bit delta-sigma modulators,” IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 40, no. 8, pp. 461-466, Aug. 1993

[15] Richard Schreier and Bo Zhang, "Delta-sigma modulators employing continuous-time circuitry," IEEE Transactions Circuits and Systems-I: Fundamental Theory and Applications, vol. 43, no. 4, pp. 324-332, Apr. 1996

[16] H. Lin, J. da Silva, B. Zhang and R. Schreier, "Multi-bit DAC with noise-shaped element mismatch," Proceedings of the IEEE International Symposium on Circuits and Systems, vol. 1, pp. 235-238, May 1996

[17] Tao Shui, Richard Schreier, and Forrest Hudson, "Mismatch-shaping DAC for lowpass and bandpass multi-bit delta-sigma modulators," Proceedings of the 1998 IEEE International Symposium on Circuits and Systems, vol. 1, pp. 352-355, Jun. 1998

[18] Zhenyong Zhang and Gabor C. Temes, "A segmented data-weighted-averaging technique," Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 481-484, May 2007

[19] Tai-Haur Kuo, Kuan-Dar Chen, and Horng-Ru Yeng, "A wideband CMOS Sigma-Delta modulator with incremental data weighted averaging," IEEE Journal of Solid-State Circuits, vol. 37, no. 1, pp. 11-17, Aug. 2002

[20] Marko Neitola, and Timo Rahkonen, "A generalized data-weighted averaging algorithm," IEEE Transactions Circuits and Systems-II: Express Briefs, vol. 57, no. 2, pp. 115-119, Feb. 2010

[21] Todd S. Kaplan, Joseph F. Jensen, Charles H. Fields, and Mau-Chung Frank Chang, "A 2GS/s 3-bit Delta-Sigma-modulated DAC with tunable bandpass mismatch shaping," IEEE Journal of Solid-State Circuits, vol. 40, no. 3, pp. 603-610, Mar. 2005

[22] Tao Shui, Richard Schreier, and Forrest Hudson, "Mismatch shaping for a current-mode multibit delta-sigma DAC," IEEE J. Solid-State Circuits, vol. 34, pp. 331-338, Mar. 1999

[23] Richard Schreier and Bo Zhang, "Noise-shaped multi-bit D/A converter employing unit elements", Electronic Letters, vol. 31, no. 20, pp.1712 -1713, Sep. 1995

[24] Nan Sun, "High-order mismatch-shaping in multibit DACs," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 58, no. 6, pp 346-350, Jun. 2011

[25] Kok Lim Chan, Nevena Rakuljic, and Ian Galton, "Segmented dynamic element matching for high-resolution digital-to-analog conversion", IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 55, no. 11, pp. 3383 – 3392, Dec. 2008

[26] Asaf Fishov, Eric Siragusa, Jared Web, Eric Fogleman., Ian Galton, "Segmented mismatch-shaping D/A conversion," IEEE International Symposium on Circuits and Systems, vol. 4, pp. 679-682, Aug. 2002

[27] Robert Adams, Khiem Q. Nguyen, and Karl Sweetland, "A 113-dB SNR oversampling DAC with segmented noise-shaped scrambling," IEEE Journal of Solid-State Circuits, vol. 33, no. 12, pp. 1871-1878, Dec. 1998

[28] Nan Sun, "High-order mismatch-shaped segmented multibit ΔΣ DACs with arbitrary unit weights," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 59, no. 2, pp. 295-304, Feb. 2012

[29] Arindam Sanyal and Nan Sun, "A simple and efficient dithering method for vector quantizer based mismatch-shaped ΔΣ DACs," IEEE Proceedings of International Symposium on Circuits and Systems, pp. 528 – 531, May 2012

[30] Heng-Yu Jian, Zhiwei Xu, and Mau-Chung Frank Chang, "Delta–Sigma D/A Converter Using Binary-Weighted Digital-to-Analog Differentiator for Second-Order Mismatch Shaping," IEEE Transactions Circuits and System II: Express Briefs, vol. 55, no. 1, pp. 6-10, Jan. 2008

[31] Akira Yasuda, Hiroshi Tanimoto, and Tetsuya Iida, "A third-order Δ-Σ modulator using second-order noise-shaping dynamic element matching," IEEE Journal of Solid-State Circuits, vol. 33, no. 12, pp. 1879-1886, Dec. 1998

[32] Jared Welz and Ian Galton, "Necessary and sufficient conditions for mismatch shaping in a general class of multibit DACs," IEEE Transactions Circuits and System II: Analog and Digital Processing, vol. 49, no. 12, pp. 748–759, Dec. 2002

Vita

Li You was born in Urumqi, Xinjiang Uyghur Autonomous Region, P. R. China on September 2, 1991. She received a Bachelor degree of Science from Peking University in 2012. In Fall 2012, Li enrolled in the Master's program in the Department of Electrical and Computer Engineering at the University of Texas at Austin. In 2013, Li joined Dr. Sun's research group at the University of Texas at Austin.

Permanent email: lyou_ece@utexas.edu

This thesis was typed by Li You.