Designing High Speed Current Steering Digital to Analog Converter Using Berkeley Analog Generator

Mohammadamin Torabi

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2019-64
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-64.html

May 17, 2019

Acknowledgement

First and foremost, my gratitude goes to my advisor Professor Ali M. Niknejad for his insightful advice and support. I also like to thank professor Nikolic for his comments, helpful feedbacks, and for reading the nal report. I would like to express my appreciation to all sta and colleagues at the BWRC for their support. I learned so much from them during my stay at Berkeley. Especially, I would like to thank students and faculty who worked on Hydra, and BAG, and helped me tremendously, Greg LaCaille, Marko Kosunen, Emily Naviasky, Zhongkai Wang, Kourosh Hakhamaneshi, Nima Baniasadi and Pengpeng Lu. It

was a privilege to learn from, and be associated with this group.

Designing High Speed Current Steering Digital to Analog Converter Using Berkeley Analog Generator

by

Amin Torabi

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ali Niknejad, Chair
Professor Borivoje Nikolic

Spring 2019

The thesis of Amin Torabi, titled Designing High Speed Current Steering Digital to Analog Converter Using Berkeley Analog Generator, is approved:

Chair ______________________________ Date ____________

______________________________ Date ____________

______________________________ Date ____________

University of California, Berkeley

Designing High Speed Current Steering Digital to Analog Converter Using Berkeley Analog Generator

Abstract

Designing High Speed Current Steering Digital to Analog Converter Using Berkeley Analog Generator

by

Amin Torabi

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Ali Niknejad, Chair

In this work, a high speed current steering Nyquist Digital to Analog Converter (DAC) is designed and developed in 16nm TSMC technology, as part of the data converter module of the Massive MIMO project. In this work, the focus is to write a layout generator using the Berkeley Analog Generator (BAG) so the design process can be captured, parameterized, and reused for future uses. This requires developing a systematic approach to data converter design: the required transistor parameters, as well as the layout process parameters can be calculated and implemented by the generator as a function of the required spec. This particular application requires an 8 bit DAC with ENOB (effective number of bits) of 6 bits working at the Nyquist rate. With a sampling rate of 5GHz and large bandwidth linearity requirements of this application, this poses various challenges for the designer.

Contents

List of Figures

List of Tables

Acknowledgments

It has been a great experience for me to study at the University of California, Berkeley. The knowledge and experience gained at Berkeley are going to benefit me along my career path. First and foremost, my gratitude goes to my advisor Professor Ali M. Niknejad for his insightful advice and support. I also like to thank professor Nikolic for his comments, helpful feedbacks, and for reading the final report. I would like to express my appreciation to all staff and colleagues at the BWRC for their support. I learned so much from them during my stay at Berkeley. Especially, I would like to thank students and faculty who worked on Hydra, and BAG, and helped me tremendously, Greg LaCaille, Marko Kosunen, Emily Naviasky, Zhongkai Wang, Kourosh Hakhamaneshi, Nima Baniasadi and Pengpeng Lu. It was a privilege to learn from, and be associated with this group.

Chapter 1

Introduction

In order to serve the growing demand for wireless data traffic, the telecommunications industry is starting to use mm-wave spectrum to exploit the very wide channel bandwidths available at high frequencies. The Hydra project is designing a massive MIMO operating at mm-wave, with a 128-antenna array serving 16 simultaneous users in the E-band (71-76 GHz). The D/A converter designed in this project is part of the custom analog front-end chips, which will enable realizing MIMO on an integrated system. Therefore the specification of this DAC are governed by the signal processing needs of Hydra.

For a high speed DAC, the layout parasitics can have a significant impact on its dynamic performance. This usually means that the layout and design will be modified after the first pass layout and extraction. Unfortunately, each iteration may need as much time as the first layout. As the processes get smaller and design rules become more complex, the design effort during this process can increase exponentially. Furthermore, analog design procedure is highly dependent on designer experience, which makes the re-use of a design much more difficult and time consuming. Berkeley Analog Generator (BAG) has tried to address this problem. In 2018, Chang et al introduced BAG2 which enables writing process-portable circuit generators [3]. Using this methodology, the designer captures the design process as an executable code with various specifications as its inputs. A layout generator can then generate a layout (draw polygons) using dynamic specifications, such as transistor width, while ensuring that the design rules are followed. This approach has been used extensively for the layout design of this DAC. Using this approach also requires the development of a set of principles and a reasonable methodology when designing high speed DACs. This requires a deep understanding of error mechanisms and proper layout techniques.

Organization

The first chapter covers the theoretical aspects of the design. This includes the common error sources and traditional ways of handling them. Next, the design approach for this application is discussed. This includes the design choices and the setup used to arrive at those conclusions. Next, the layout design is discussed in detail, including the choices and consideration throughout the process. Finally the conclusions and potential future work for this project is presented.

Chapter 2

DAC fundamentals

This chapter deals with the basics of DAC design, including the metrics, architecture, and specifications for this particular applications.

Specifications

For this application, the specification is as follows:

Resolution	8 bits
ENOB	6 bits
SFDR	50 dB
VDD	900 mV
Max fs	5GHz
Max bandwidth	2.5 GHz
Max Swing	300 mV

Table 2.1: DAC specifications

While minimizing the power consumption and area. Furthermore, this DAC needs to have 3 bits of gain control, therefore the full scale voltage can vary from 37.5mV to 300mV. [1] The load of the DAC are 50Ω resistors.

[1] this is 600 mV differential peak to peak voltage

Architecture

For this application, the focus is on Current Steering topology. This topology ensures that the fast sampling frequency can be maintained. Furthermore, Current Steering architecture does not require any buffers and it does not have a high load. This makes this the topology of choice for high speed DACs [10].

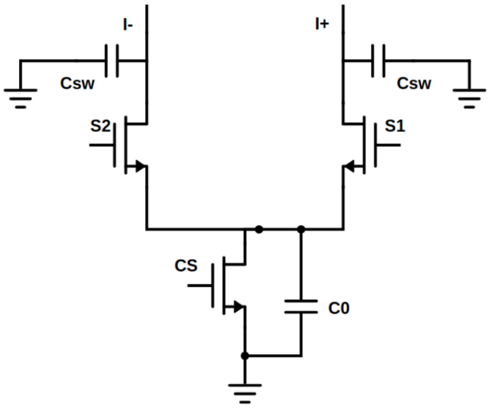


Figure 2.1: Basic Current Steering Cell

In this topology, CS is the main current source which will generate the least significant bit (LSB) current. S1 and S2 are minimum sized switching devices which turn on and off to steer the current to left or right.

2.1 Performance Metrics

In order to understand the basics of designing a DAC and its trade-offs, the various metrics for evaluating a DAC need to be explored. But first, it is important to discuss the response of an ideal DAC.

Ideal DAC response

An ideal non-return to zero (NRZ) DAC response essentially acts as a zero-order sample and hold block: it will output the same value until the input has changed by 1 LSB. This effectively corresponds to a sinc function in frequency domain:

$$|H(2\pi f_{in})| = \frac{sin(\pi f_{in}/f_s)}{\pi f_{in}}$$

This implies that for $f_{in} = f_s$ the magnitude is zero. However, this also implies that as the input frequency increases, the signal will get attenuated due to sinc roll-off.

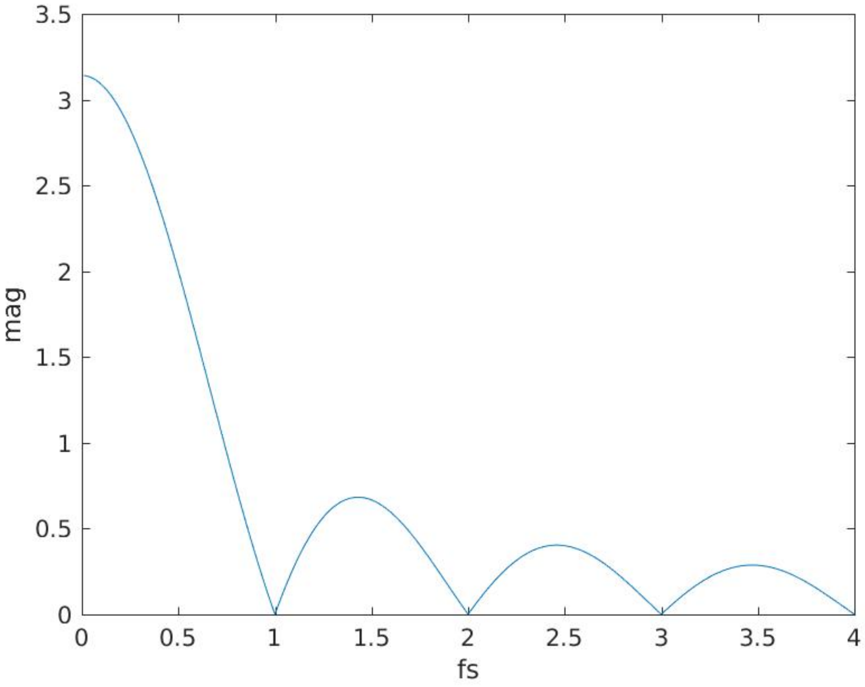


Figure 2.2: Ideal DAC frequency response

For instance, for the Nyquist case, the signal gets attenuated by -3.92 dB. Furthermore, this implies that the DAC images and harmonics at higher frequencies will not be attenuated. In order to compensate for this, DAC designers either use over-sampling techniques where $f_{in} < f_s/2$, or use different switching mechanism (such as return to zero DACs) with various pulse widths[4]. Furthermore, there is usually a low-pass filter at the output to attenuate the images and return a smooth signal. However, for this application, we proceed with the simple NRZ architecture as it meets the specifications.

Static Metrics

There are two important static metrics for DACs. Differential nonlinearity (DNL) and integral nonlinearity (INL). DNL refers to the difference between two adjacent output levels. For instance if two output levels are 1.5 LSB apart, DNL for that code is 0.5 LSB. In order to insure the DAC output increases monotonically, DNL should not be larger than 1 LSB. INL is the measure of actual output level minus the ideal output level. To measure INL, a ramp input is given to the DAC, then a line from zero to full scale is drawn and effective LSB calculated. Then for each output level, the distance from that ideal line and the output level is measured and normalized by effective LSB.

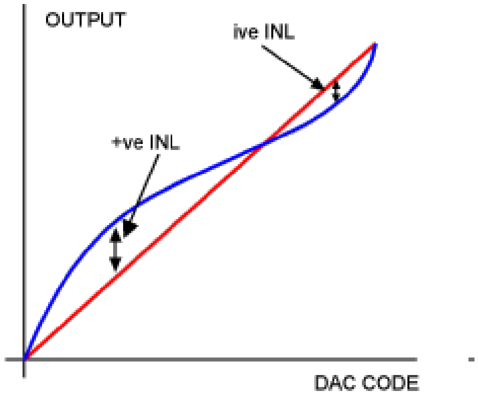


Figure 2.3: INL measurement

The Static performances of the DAC are fundamentally not affected by changing the input frequency or the sampling rate. This implies for low speed DACs, they are the main source of error and nonlinearity. However, at higher frequencies, other dynamic error sources become dominant, which will be discussed.

Dynamic Metrics

There are several important dynamic metrics for the DAC. They are measured by exciting the DAC by the codes corresponding to a single sine wave, and measuring the output using a spectrum analyser.

- **Signal to Noise Ratio (SNR)**: This is the spectral power of the input tone (fundamental) over the noise floor. For an ideal DAC, the noise floor is dominated by the quantization noise, which is a direct function of resolution. For a sinusoidal input, the RMS quantization noise can be expressed as [10]:

  $$Q_{RMS} = \frac{I_{LSB}}{\sqrt{12}}$$

  and since full scale power is $\frac{2^N(I_{LSB})}{2\sqrt{2}}$,

  $$SNR = \frac{I_{LSB}/\sqrt{12}}{2^N(I_{LSB})/2\sqrt{2}}$$

  Simplifying and taking the log we have:

  $$SNR = 6.02N + 1.76$$

  where N is the resolution in number of bits.

- **Spurious Free Dynamic Range (SFDR)**: This is the measure of signal power over the largest spur. Usually spurs are data dependent, and thus the largest spurs are the distortion elements, such as the 2nd or 3rd harmonic. For high speed DACs, SFDR is usually above the quantization noise floor.

- **Signal to Noise & Distortion Ratio (SNDR)**: This is the measure of signal power over the sum of all other noises (including the distortion and quantization noise). For high speed DACs, SNDR is usually limited by SFDR, since harmonics are above the noise floor.

- **Effective Number of Bits (ENOB)**: this is essentially the effective number of bits, which includes taking into account all noise sources. It is defined as:

$$ENOB = \frac{SNDR - 1.76}{6.02}$$

Note that it is possible to define other dynamic metrics, for instance, exciting the DAC by more than 2 tones and measuring the intermodulated distortion.

2.2 Sources of Error

There are two main types of errors present in DACs: one is the static type, which is independent of clock frequency, and the other is the dynamic type, which varies as the frequency of the data or the clock changes. Usually the overal DAC performance is dominated by the latter for high speed DACs. It is also important to note that there are fundamental trade offs between the two types which will be discussed here.

Static sources

Random mismatch

Current sources random mismatch sets an upper limit to the DAC INL and thus linearity. This mismatch is due to the unavoidable statistical variation for each transistor and process parameters. Although the exact way the mismatch occurs is highly process dependent, Pelgrom model [11] gives us a first order understanding of it:

$$\sigma^2\left(\frac{\Delta I}{I}\right) = \frac{1}{WL}\left[A_\beta + \frac{4A_{VT}}{(V_{GS} - V_T)^2}\right]$$

In this model, A_β and A_{VT} are process parameters, corresponding to the current factor and threshold voltage variation. This equation implies that increasing the area of the transistor improves matching. Also this implies that increasing the overdrive voltage improves the current accuracy since the threshold voltage variation will have less of an effect on the current output. It is also worth noting that the mismatch impacts the switch drivers as well: an imbalance between the strength of inverters driving the switches translates to an error at the output.

IR drops in layout

For a large DAC, there could be some variation on the ground node of the transistors due to some IR drop. This effect can introduce a gradient and thus current variation for each unit cell.

Finite output resistance

Since each cell has finite resistance, the overall resistance of the DAC can vary as a function of the input code. This means that the output voltage deviates from the ideal case due to this data dependency and thus generates non-linearity.

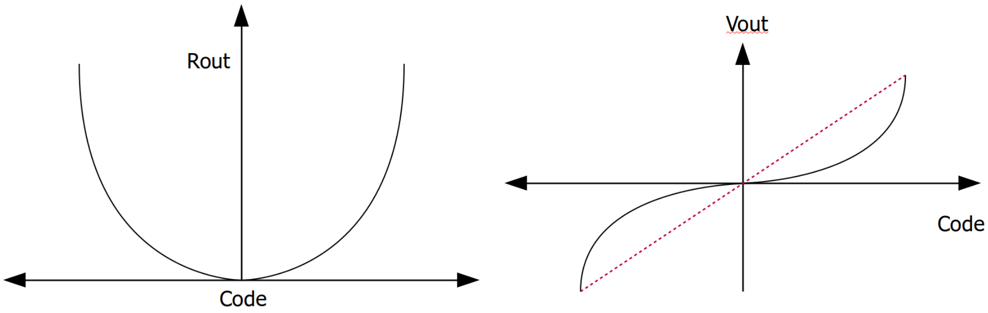


Figure 2.4: R_{out} vs input code and V_{out} vs code

For a differential DAC, this mechanism results in 3rd order harmonic distortion [5]:

$$HD_3 = \left[\frac{NR_L}{4\Delta R}\right]$$

Where N is the number of bits. This is a very important distortion mechanism and will be covered more in detail.

Dynamic Sources

Dynamic sources are usually dominant for a high frequency DAC. There are many mechanism which contribute to non linearity, and here is a summary of the relevant ones.

Finite output impedance

At high frequencies, the DAC will have an output impedance (instead of only resistance), which depends on the operating frequency. The output impedance of each unit cell is therefore limited and degrades as the frequency increases. This contributes to 3^{rd} order nonlinearity, the same way finite output resistance contributes.

Below is the small signal model of a single cell where switches are in saturation:

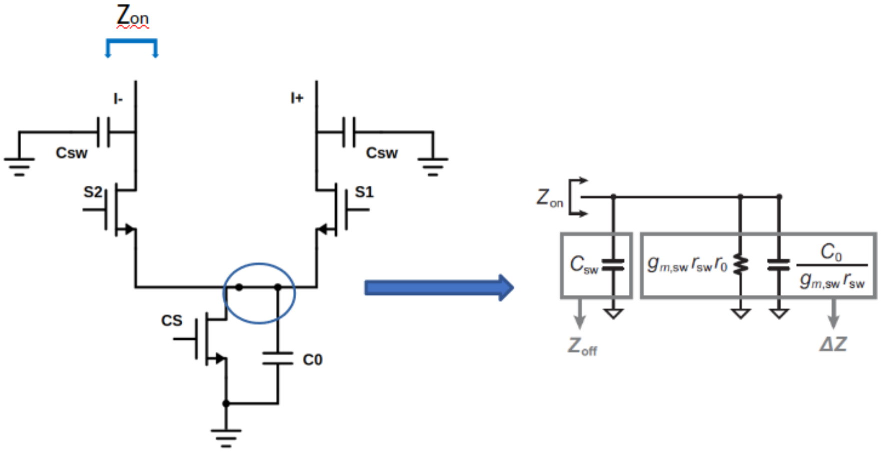


Figure 2.5: Small signal model of the unit Cell

Now consider the switching capacitance or ΔZ (this was previously ΔR), which is now a function of the drain capacitance of the current source. This capacitance causes the impedance to drop at high frequencies (after $1/2\pi r_0 C_0$) by 20dB per decade: This implies

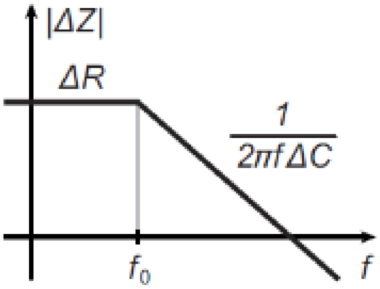


Figure 2.6: Output impedance of a Cell as a function of operating frequency

that the previous equation is now:

$$HD_3 = \left[\frac{NR_L}{4\Delta Z}\right]$$

This reveals a fundamental tradeoff between static and dynamic non-linearity: in order to lower the impact of random mismatch, we need large current sources. However at higher frequencies, the drain capacitance degrades the impedance, so we need smaller current sources. In other words, for good dynamic performance, we may need to sacrifice the static performance. This is one of the reasons that sophisticated calibration loops (including using a high resolution but slow ADC) are sometimes used [2] to decouple the dynamic and static performances, specially at higher resolutions.

Code Dependent switching constant variation

For a current source, the output capacitance is constantly changing with the input code. At each transition, the switching element sees a slightly different capacitance, and thus will have a different time constant. This in turn contributes to second order harmonic distortion[12].

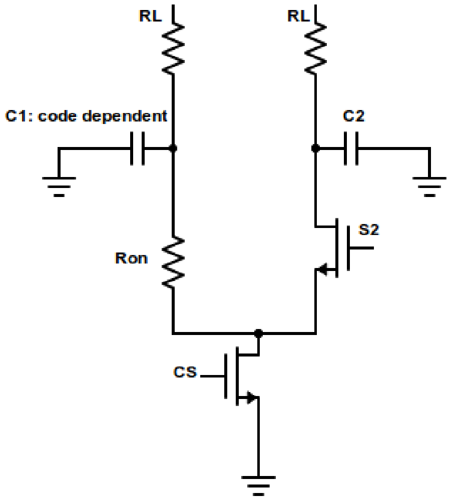


Figure 2.7: $R_{out}C1$ (Switch time constant) and its code dependence.

Code Dependent switching capacitance

During each switching action, the circuit is transitioning from one side to the other (and thus the term Current Steering). There exists a ΔC associated with this switching action which is the switching part of the ON capacitance. The voltage across this ΔC is code dependent. For instance, when a cell switches from positive to negative, the charge difference is $\Delta Q = \Delta C V_{out}$. This ΔQ can be compensated by an error current ΔI, which will cause distortion.

This mechanism also causes a third order distortion [5].

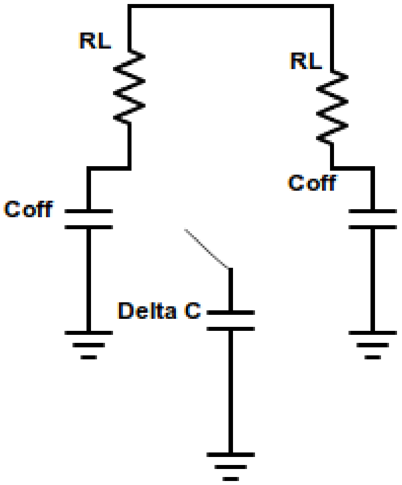


Figure 2.8: Switching capacitance at the moment of transition.

Jitter

Clock jitter also contributes to error at the output. A timing error in the sampling phase can cause an amplitude error at the output.

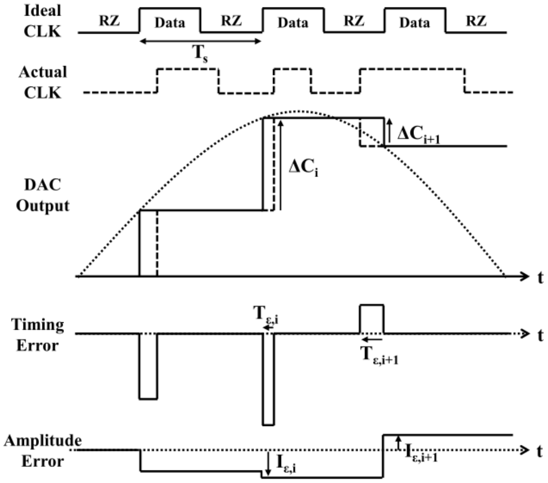


Figure 2.9: Timing error conversion to amplitude error.

Clock jitter poses a fundamental limit to the achievable DAC resolution. It also impacts Non Return to Zero and Return to Zero DACs differently. For instance, for a Return to Zero DAC, the timing error will also be a function of sampling frequency, since the output goes back to zero, at each sampling frequency regardless of the data. However, for a Non Return to Zero DAC, the error is a function of the input data. The SNR limit for a NRZ DAC can be calculated [7]:

$$SNR_{NRZ} = \frac{1}{4\sigma^2\pi^2 f_{in}^2}$$

Where f_{in} is the frequency of the input data and σ is the clock jitter in seconds.

Chapter 3

Design Approach

Now that we have developed some theoretical understanding on the error sources present in a DAC, we can develop a methodology for design. This section deals with the design procedure of the DAC, and its systematic testing.

3.1 Segmentation

The first decision to be made when designing a DAC is the segmentation between binary and thermometer bits. Thermometer coding refers to a structure where each switching element generates equal current, whereas the binary coding refers to a structure where the switching elements generate currents with powers of 2 (eg: 1x, 2x, 4x etc). Each coding structure has advantages and disadvantages which are briefly listed below:

- **Area:** A thermometer coded DAC occupies a larger area as there more switching elements. For a N bit DAC, 2^N units are needed, where as a binary coded DAC uses $2N$ cells. Larger cells mean more difficult routing and clock distribution. Thermometer coding also requires a digital binary to thermometer converter.
- **Glitches:** Binary coded DACs generate more glitches at the output. Consider a change from 011 to 100: two cells (1x and 2x) need to turn off, and a large cell (4x) needs

to turn on. This introduces more glitches in comparison with thermometer structure where only only one unit is turning on to transition from 011 to 100. The glitches produced with binary codes are data dependent and unlike thermometer code glitches, might contribute to nonlinearity [9].

- **Monotonicity** Thermometer DACs are fundamentally monotonic as the output can only go up as the input code is increased. However, for binary encoding, this condition is harder to meet: consider again changing a code from 0111 to 1000 (major bit transition). For monotonicity, we need to ensure that the sum of 1x, 2x and 4x cell is actually less than the 8x cell to within 0.5 LSB. This is challenging due to the spread of statistical mismatch (theoretically impossible to guarantee).

Impact of Mismatch

DAC segmentation has major impact on matching consideration. A converter is considered to have N bits of resolution if $|DNL| < 0.5$ LSB for every code, since this implies that the error is within the bounds of quantization error. Furthermore, a DAC output is monotonic if $|INL| < 0.5$ LSB. However, the DNL of a binary weighted DAC varies much more significantly with mismatch. Suppose the current generated by a cell has a variance of σ_{lsb}^2. It is possible to observe that[8]:

$$\sigma_{DNLMax}^2 = B\sigma_{lsb}^2$$

Where B is the number of maximum number of current sources in a transition. In other words, the DNL for a binary weighted DAC will be worst at the mid-code transition, where 2^N cells are switching. Therefore:

$$\sigma_{DNLmax} = 2^{N/2}\sigma_{lsb}$$

On the other hand for a thermometer coded dac, there is only one cell that is switching, thus:

$$\sigma_{DNLmax} = \sigma_{lsb}$$

This shows the clear advantage of thermometer coding when matching is considered for DNL. On the other hand, thermometer or binary coding, does not impact the INL. This is intuitive since INL stands for the integrated measure, and therefore the sum of statistical variations to each code must be the same.

Segmentation and area trade-off

Assuming area is a constraint, it is possible to observe the following trade off [9]: The idea

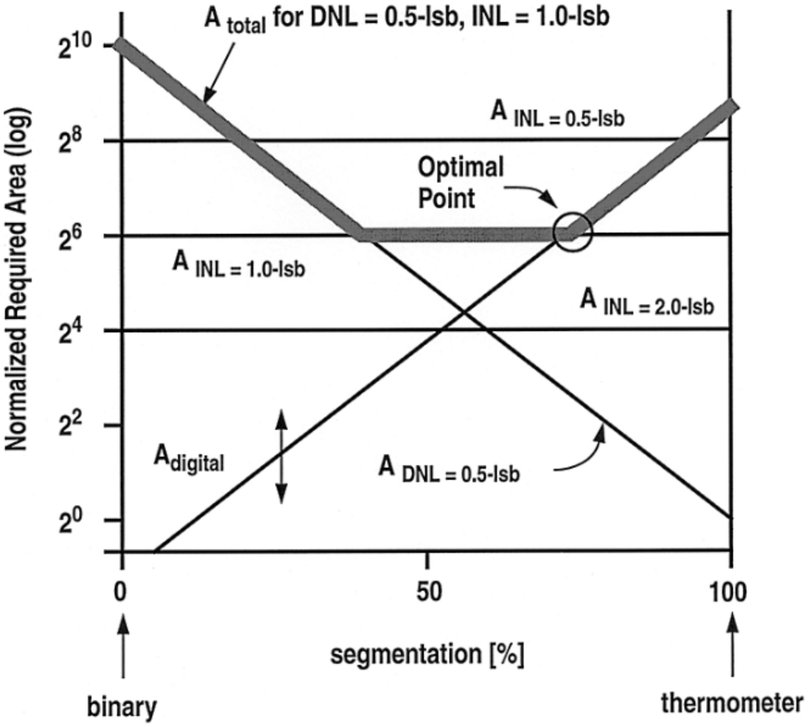


Figure 3.1: Area versus segmentation trade off proposed by Lin [9]

here is that as the segmentation moves from fully binary encoding to thermometer encoding (0% to 100%), the digital area required for routing, switching and binary to thermo conversion increases. On the other hand, the normalized analog area (which is the current source area) required for matching decreases. For any technology and resolution, there exists an analog area where the INL per code is within 1 LSB. The optimum segmentation is where the “digital area” intersects the minimum “analog area” required to meet the INL spec.

Segmentation choice

The choice to segment the DAC depends on the application, area constraint, switching power constraint, linearity and the resolution. For instance, for a high resolution DAC ($N > 10$), it becomes very difficult to have more than 6 bits of thermometer coding. Very high resolution DAC use complex calibration techniques to achieve those resolutions anyway, so the DNL errors caused by binary mismatch is less problematic. If switching power is a limiting factor, more bits need to be binary coded, since that lowers the number of switching elements and digital circuitry.

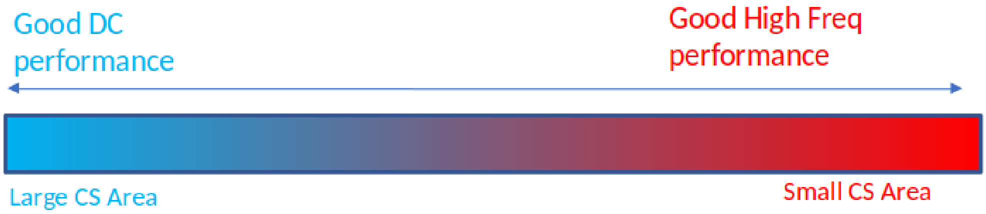
For this application, the area and the switching power is not a particular constrain. In fact the unit cell area is dictated by the power grid to make it easy to integrate to the rest of the chip (this will be discussed more in the next chapter). On the other hand, working on advanced technology nodes mean the “digital area” is actually very small. The latches and flip flops can be tiny in comparison with the current source. Newer technologies also have worst matching characteristics and lower allowable channel length. With a medium range resolution of 8 bits, this implies that the optimum segmentation has to have more thermometer codes than binary. Considering, all the above points, for this application, the segmentation is chosen to be 3 bit binary and 5 bit thermometer. However, this segmentation choice is technology dependent.

3.2 Device Sizing

There are two main trade-offs for device sizing:

1. Larger devices, improve matching, but degrade the high frequency performance due to the large capacitance at the drain of the current source.

2. Large overdrive voltage helps matching, but consumes more headroom.



For this particular application, the full scale current can be determined:

$$I_{lsb} = \frac{37.5\,\mathrm{mV}}{50\Omega \times (2^8 - 1)} = 3\,\mu\mathrm{A}$$

Furthermore, the DAC is to have gain control, and therefore the DAC load is a PMOS current mirror. This implies that the headroom is very limited. For a general purpose DAC these elements could be used as design parameter as well. However, in this application, the implication is that we only need to size devices to ensure that INL and DNL specification are met, with the smallest possible current source area while consuming the minimum headroom. The schematic of the proposed system is described in figure 3.2.
Note that the switches (S1 and S2) need to remain in saturation. Furthermore, each DAC cell has its own biasing which is not shown. The NMOS current sources need to produce a large enough current such that the PMOS load does not exhibit nonlinear behavior. Using the PMOS load also means that the DAC output is filtered by the mirror poles and the load impedance is low. Assuming sufficient bias current in the loads, this could potentially aid the high frequency performance.

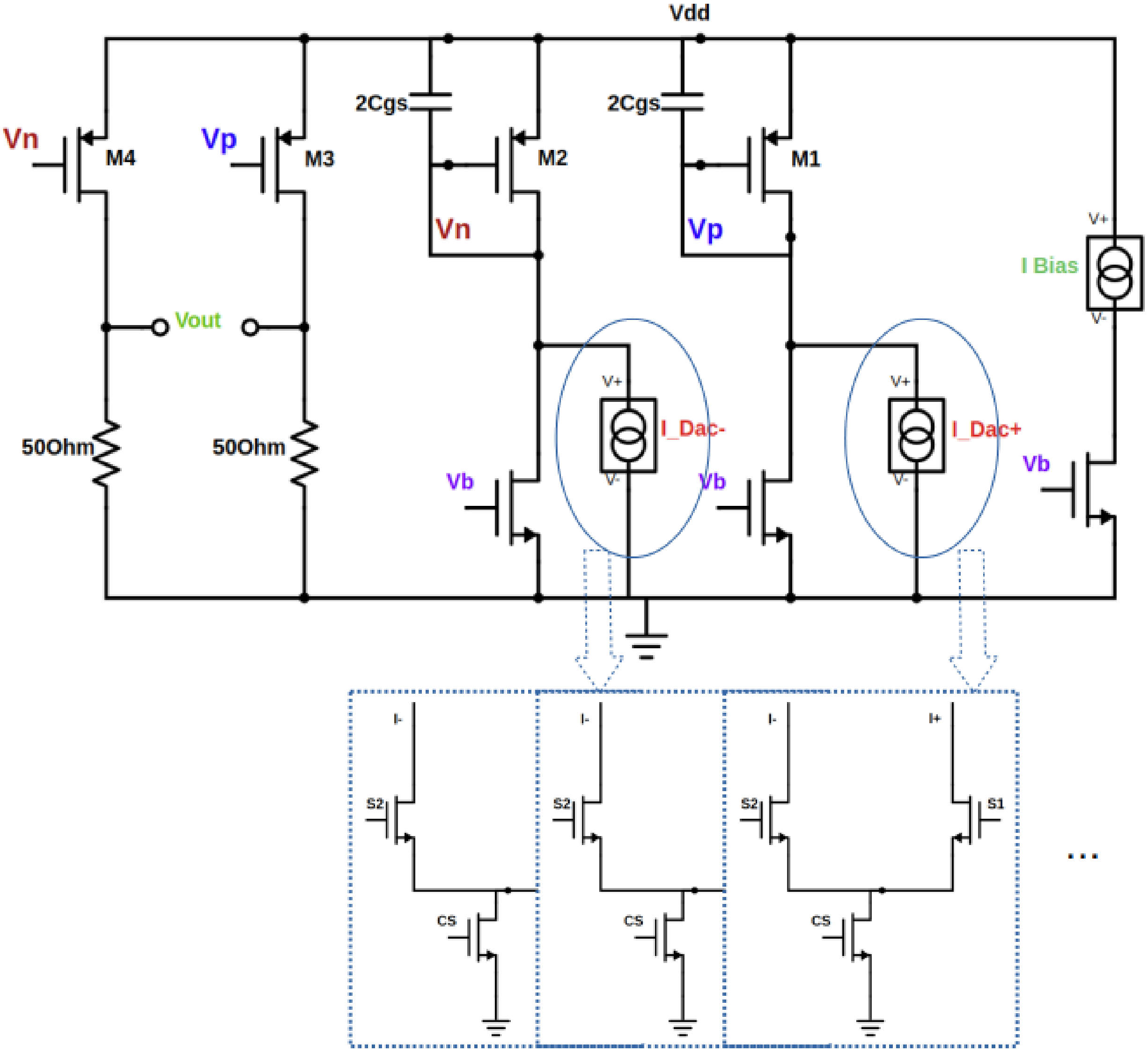


Figure 3.2: The entire system including the DACs and the PMOS diode loads and current mirrors.

Test Bench for Sizing

The proposed test bench used for finding the current source devices is depicted in figure 3.3. A Monte Carlo simulation is done where the generated DC current is measured and its variance σ^2 is found. This σ is used in a Matlab model with our ideal segmentation. This Matlab model simulates a hypothetical segmented DAC with a ramp signal and measures the INL and DNL and their rms[1]. Below is the INL DNL measured using the σ obtained from Monte Carlo simulation:

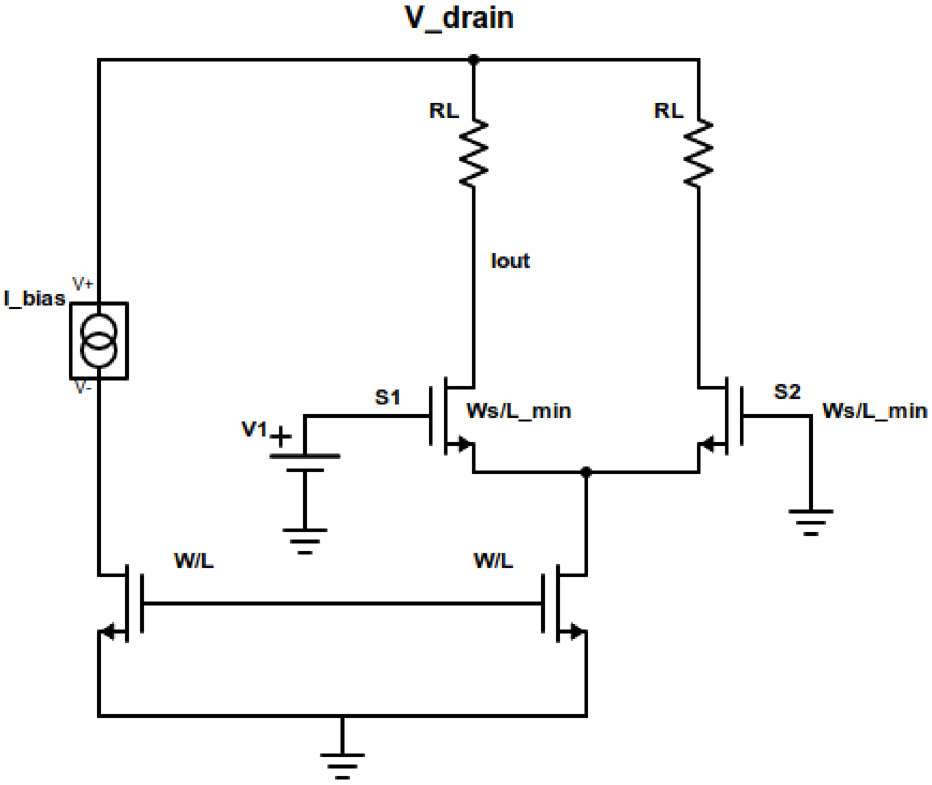


Figure 3.3: Proposed test bench where I_{out} is measured.

[1] Note that the Matlab model was initially used because simulating many ramp signal using a transient analysis would be time consuming. Also since we are measuring static errors only due to mismatch, there was no particular need to use special circuit solvers. However it is perfectly possible to incorporate this setup in BAG framework and use the results to size the transistors.

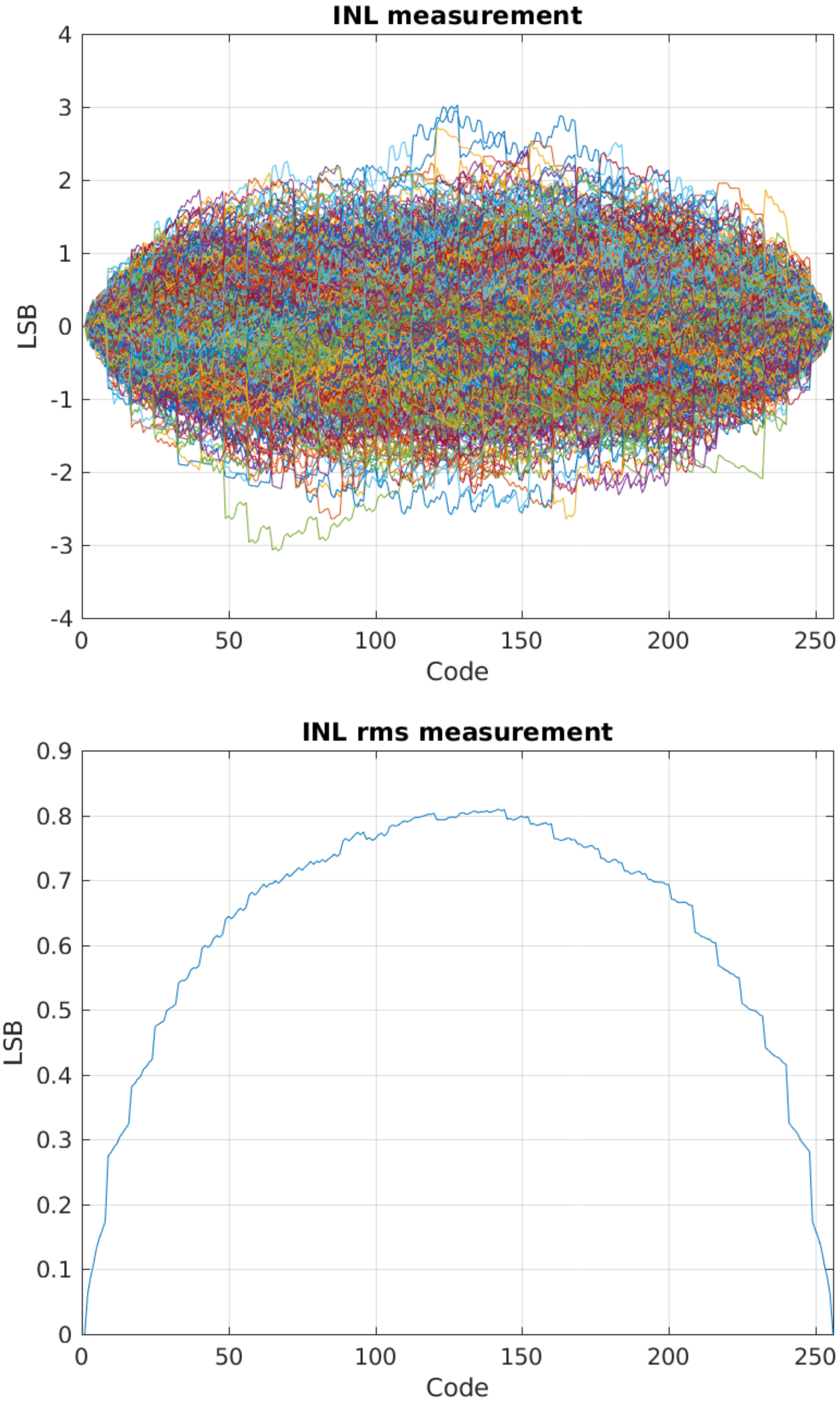


Figure 3.4: INL measured with 1000 simulations.

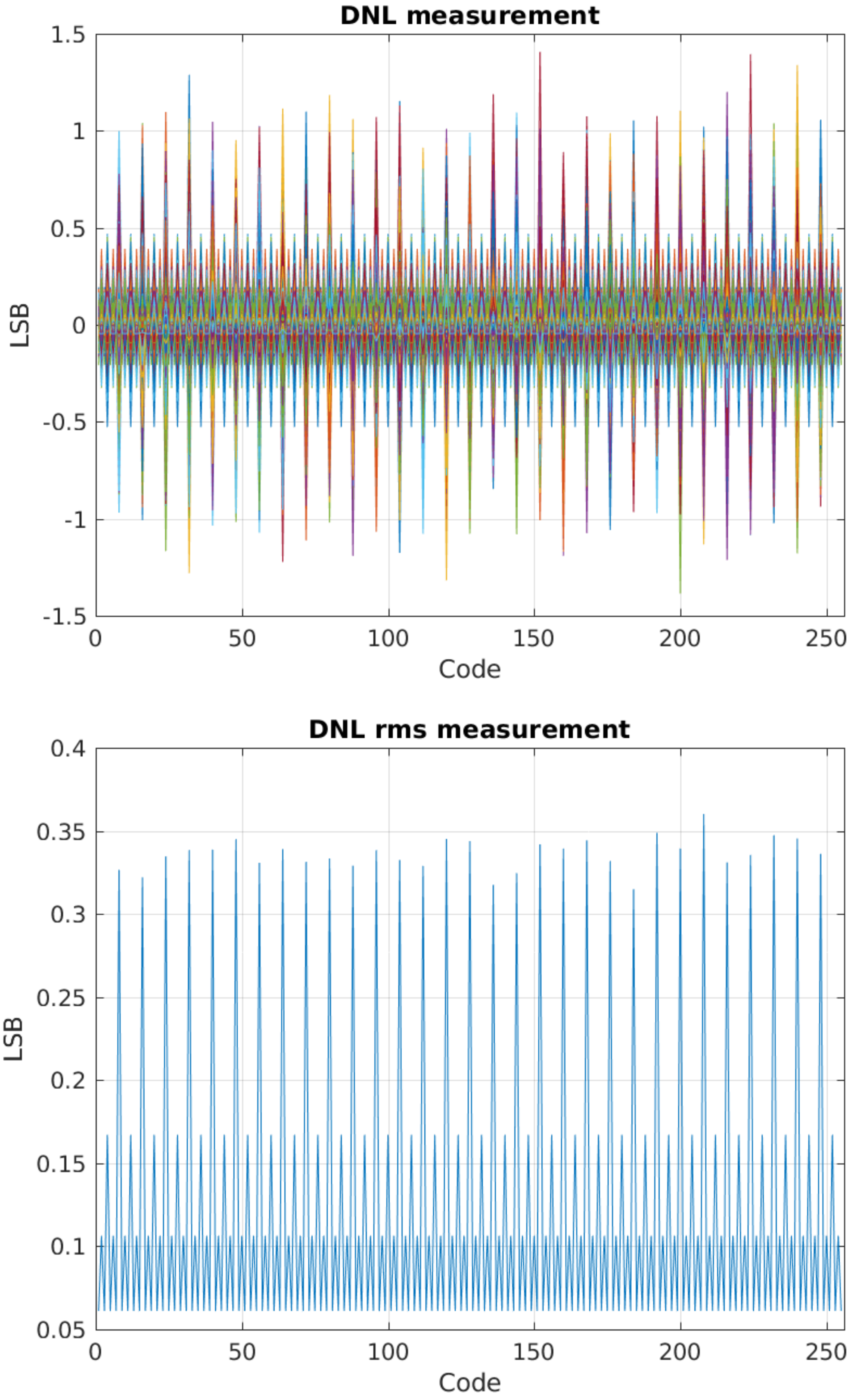


Figure 3.5: INL measured with 1000 simulations

As we see, the DNL spikes at every 4th code, which is the ”major” bit transition in the segmented DAC. In other words, the simulation confirms the theory described above. The variance of INL is also at 0.77 LSB. This sets an upper bound for low-frequency SFDR [8] as $20log(\frac{256}{0.8}) = 50.4dB$ which is below the quantization noise level.

This testbench setup can be used to ensure that the devices are large enough and INL, DNL specifications are met. Once the unit cell is properly designed, we can perform a transient analysis and confirm the results at the schematic level.

There are few more points regarding sizing that are worth mentioning:

- Minimum sized switching devices are chosen. This is to ensure the capacitance at the output is minimized.
- The switching devices are chosen to be high threshold flavor. This is to ensure that they remain in saturation when they are ON.
- For newer technologies, the maximum length may not be enough to realize the required matching. In such cases, larger lengths can be achieved by stacking the current sources in series.

Transient Simulation Test Bench

From the previous test bench, the basic unit cell and its device sizing have been found. The next step is to setup a test bench to measure the dynamic and static performance of the DAC.

The figure below shows the general test bench used to verify the performance at both schematic and post-layout levels.

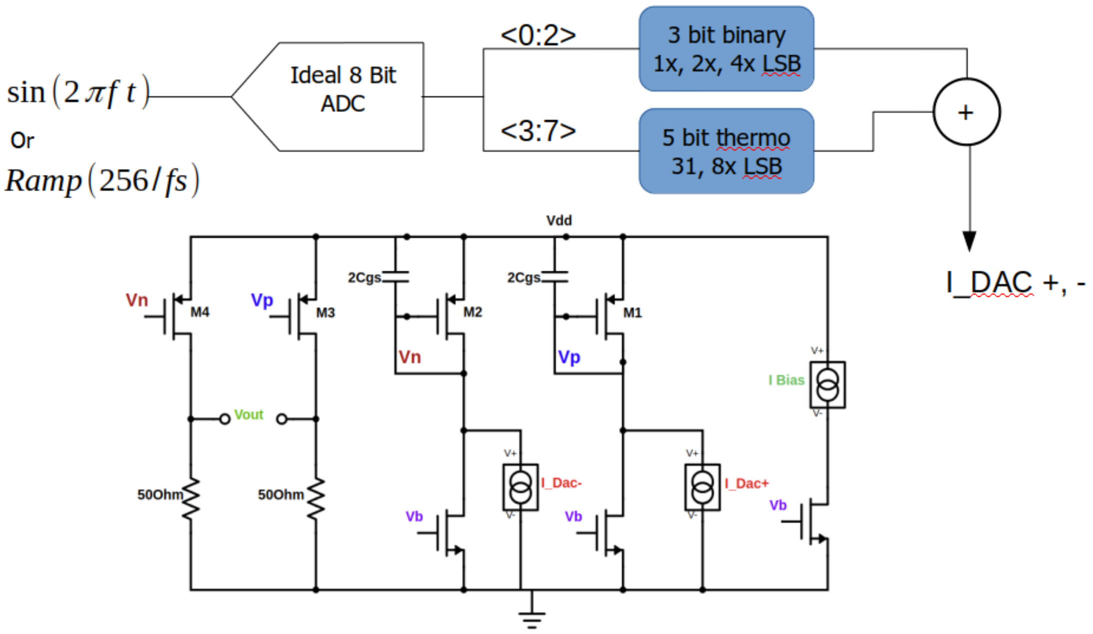


Figure 3.6: General test bench

In this figure the frequency of the sine function (f_{in}) is chosen such that:

$$fs \times \frac{M}{2^n} = f_{in}$$

Where fs is the sampling frequency (which according to specifications is at most 5GHz), 2^n is the number of samples in DFT, and M is a prime number. On the other hand the Ramp function is just a line that spans all the codes of the DAC.

The output is captured and analysed according to the type of analysis. For a dynamic performance analysis, a DFT is taken from the sinusoidal output, and for a static performance, the INL and DNL of the ramp is calculated.

3.3 Transient Simulation Test Results

A Monte Carlo simulation with 50 samples is performed over the transistor level design. The input tone is a 1.9GHz signal where the sampling frequency is 5GHz. Then the DFT of the output is taken. Below is a typical spectrum, and the statistical variation os SFDR and SNDR. As we see, the test results indicate that the DAC is meeting the spectrum with a typical ENOB of $\frac{45.2 - 1.76}{6.02} = 7.2$. However, the ENOB is not limited by harmonic distortions and spurs as the SFDR is 6dB below the SNDR. For the static measurement, the same setup is used, except the input is a ramp. A Monte Carlo simulation with 200 samples is performed and the RMS results for INL and DNL are shown below.

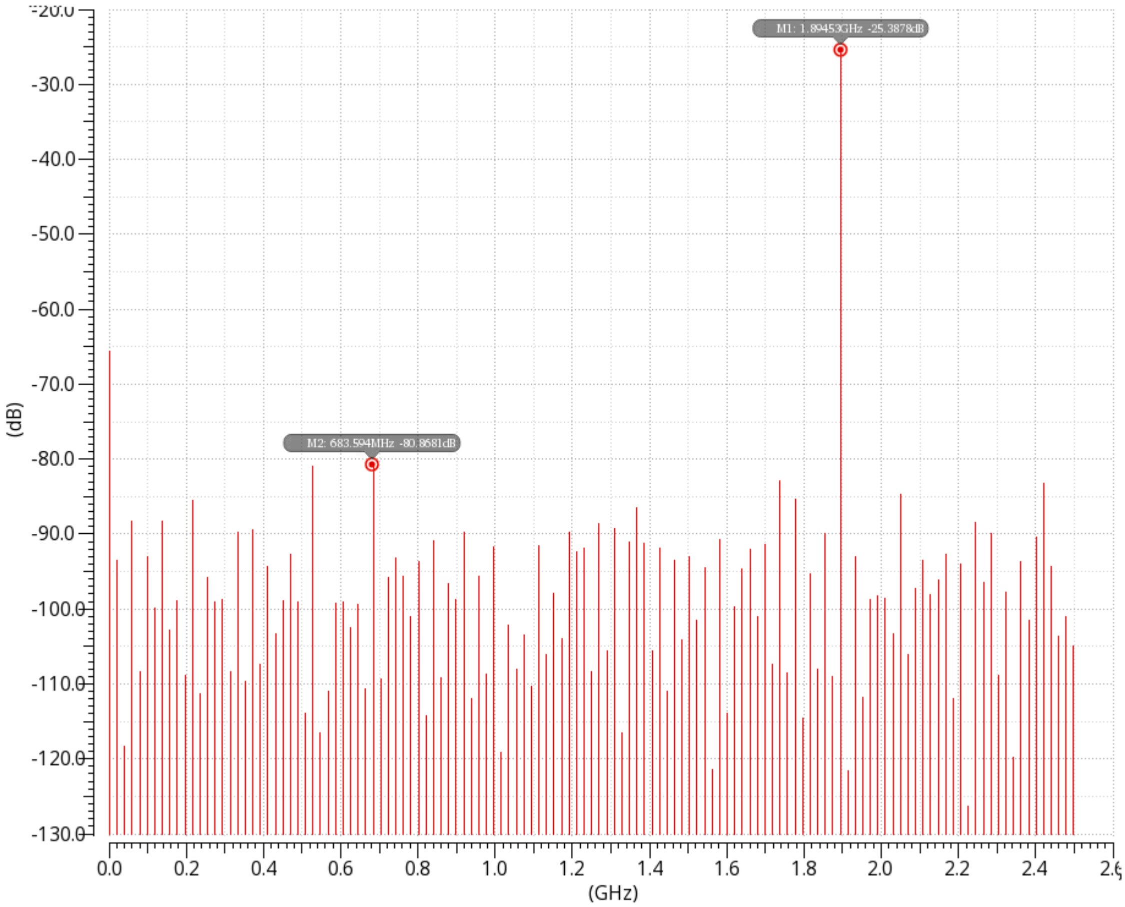


Figure 3.7: DFT of the output with 256 samples.

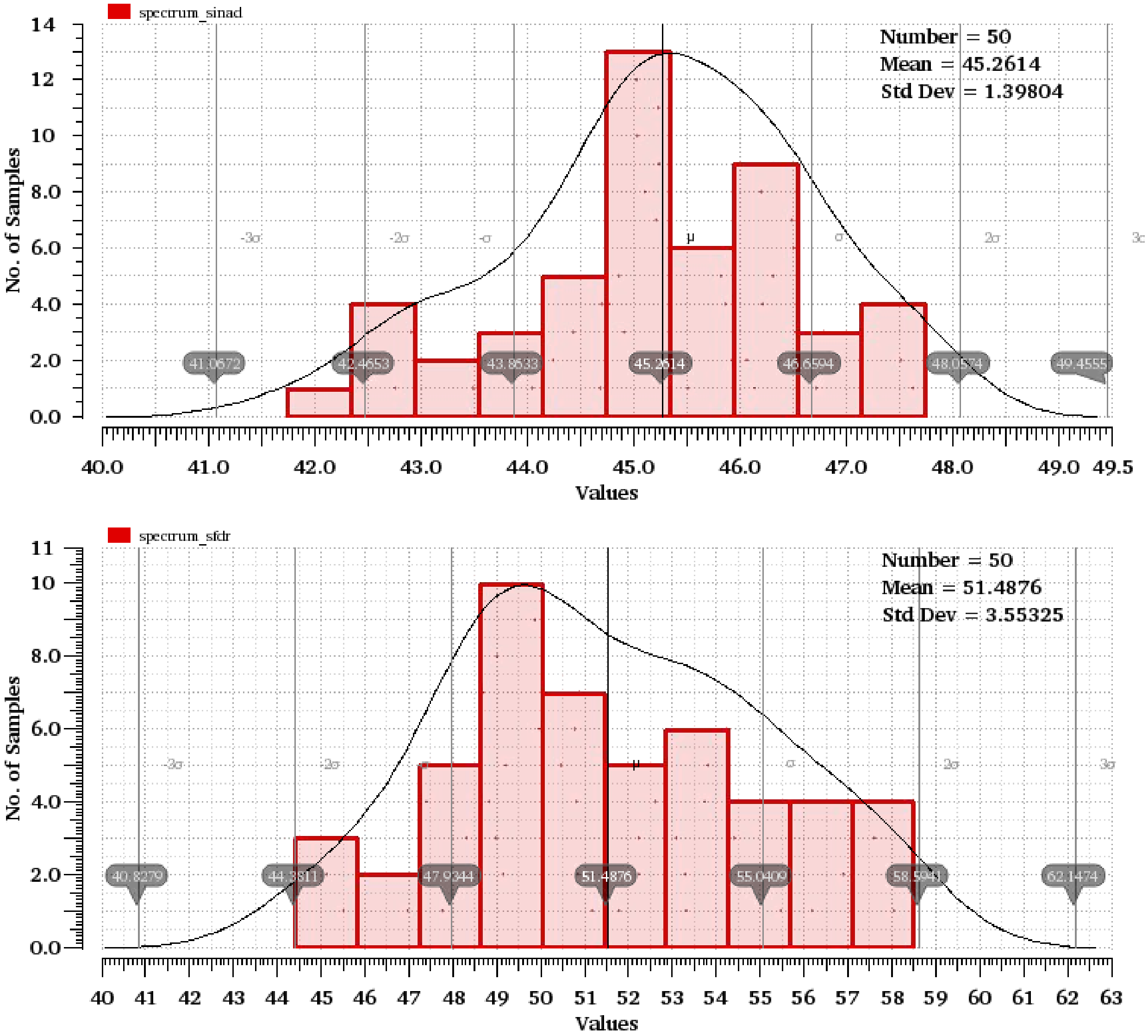


Figure 3.8: SNDR and SFDR statistical variations.

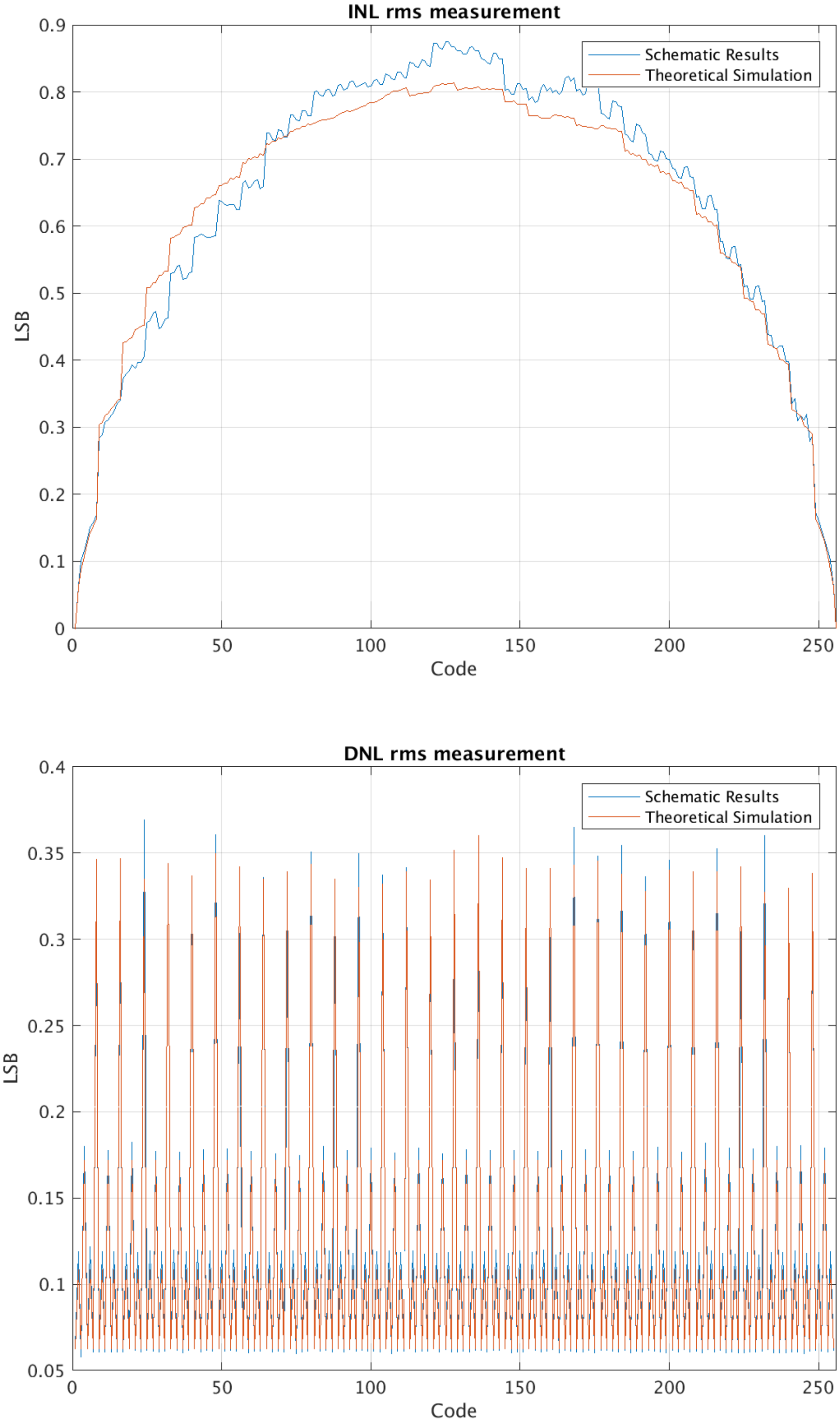


Figure 3.9: RMS of INL and DNL variation of transient simulation.

Comments and observations

It seems that the theory and measurements are in agreement and the specs are met:

- Mean SFDR is about 51dB. This is the theoretical upper bound obtained from a INL of 0.8 LSB.
- The expected RMS of INL variation obtained from the test bench matches the transient simulation performed on transistor level design. This confirms the viability of the above testbech in design procedure.
- SNDR mean at the highest frequency of operation is measured to be about 45 dB. This is about 5dB less than the ideal SNDR $(8 \times 6.02 + 1.76dB)$.

Chapter 4

Layout Design

After schematic verification of the design, the next step is to layout the design and re-test it. There are several considerations that are crucial when designing the floor plan:

- **Matching**: Layout directly impacts matching. Matching is extremely important both for static performances and high frequency applications. For instance, mismatch can increase the noise floor, or generate second and third order harmonics [1].

- **Timing**: For a DAC to produce the "right" output, all unit Cell need to switch at the same time. This implies that any time variation of the clocks or the data going to the clocks will clearly cause errors at the DAC output. For a DAC that operates at high frequency, this is a serious challenge and an upper limit to how fast the DAC can operate.

- **Interference and coupling**: When high frequency digital signals are routed together, extreme care is needed to ensure that they do not overlap and couple together. For example, consider the digital data signals (which can be very fast, roughly 1-2 GHz) can couple into the bias node of the DAC current source and cause a current variation at the output. This mechanism can cause serious disparity between post layout and simulation results.

- **Ground and supply variation**: The ground and supply could vary due to the ohmic resistance of the wiring metal, or the noise from the digital circuitry. For instance consider the noise from digital flip flops on the supply. That noise could in theory change the current output of a current source, causing an error at the output. Also, if such a noise is somehow data dependent, this could contribute to distortion.

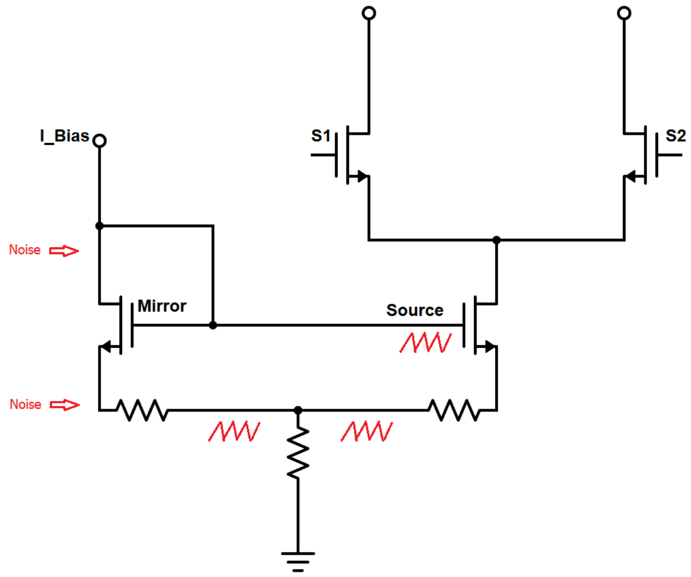


Figure 4.1: Mechanism of noise coupling into the output.

Depending on the application, each one of the above issues may be the dominant. For instance, the coupling issue and timing variation may be worst at higher frequencies.
The following section describes the methodology used for layout. In the process, the steps taken to combat each of the above issues are also discussed.

4.1 Unit Cell Design

The first step of layout is designing a unit cell. A unit cell in this case is comprised of a current mirror and two switches which are being driven by a differential flip-flop. The flip flops are necessary to ensure that all cells switch simultaneously, regardless of the delays in the data line.

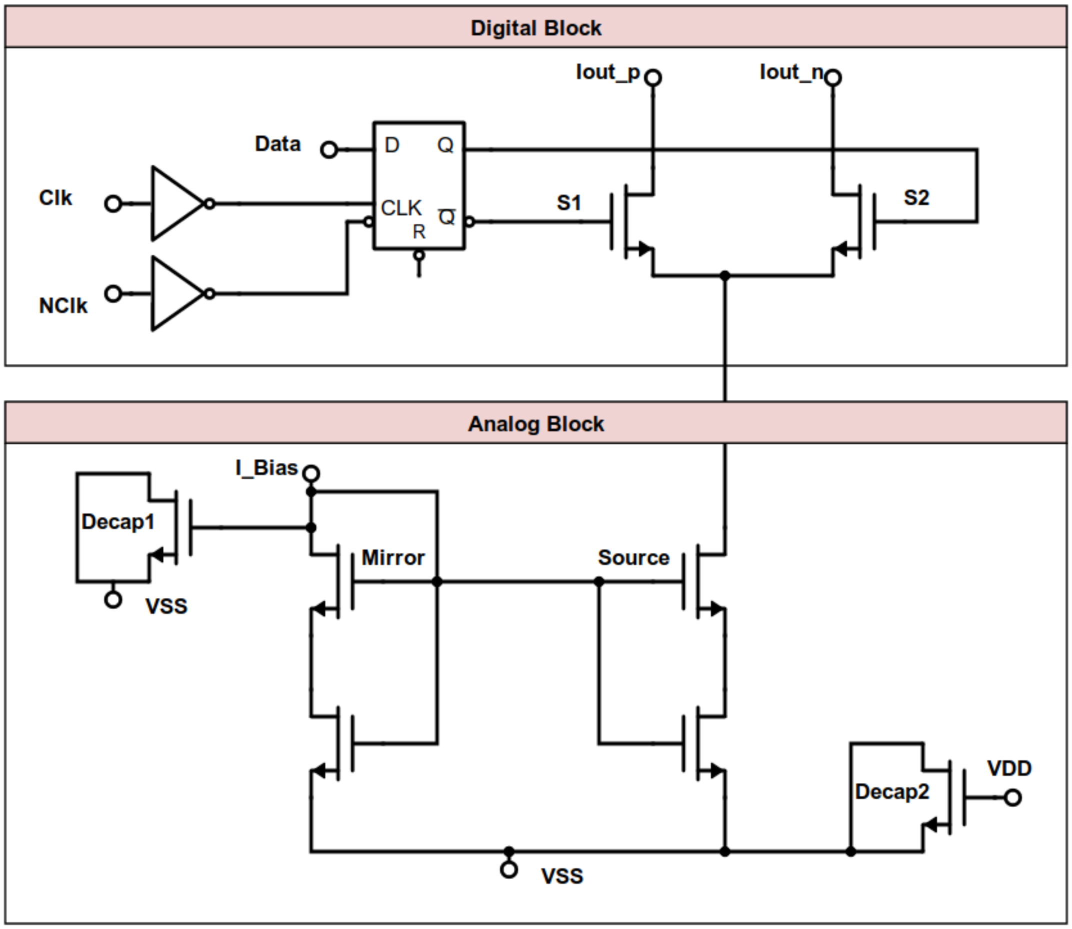


Figure 4.2: Unit Cell composed of 2 sub blocks (Analogbase instances).

Each unit cell is designed using two other sub blocks, each an Analogbase instance [1].

[1] Analogbase is the most basic building block of the BAG framework. Analogbase instances can be combined in a hierarchical manner to make a Templatebase instance.

Current Source and mirror

The current source block includes a mirror and a source transistor. Notice that in this case, since the maximum transistor length was not enough to satisfy matching requirement, the current sources were stacked to increase the length.

Since this block has relatively large transistors and many dummy transistors, this is also used to generate dummy capacitance to help with ground bouncing and the noise at the gate of the diode. Below is a snapshot of the layout. The active area is in red and the Green are the MOS capacitor and dummies. For this particular application, the length of the cell was constrained to have enough width for supply and ground connection to the rest of the chip. In other words, regardless of the current source active area size, enough dummies needed to be added to realize a certain size. Furthermore, for matching purposes, the active area needs to be surrounded by dummies. Therefore the top and bottom rows were utilized as high density dummy caps.

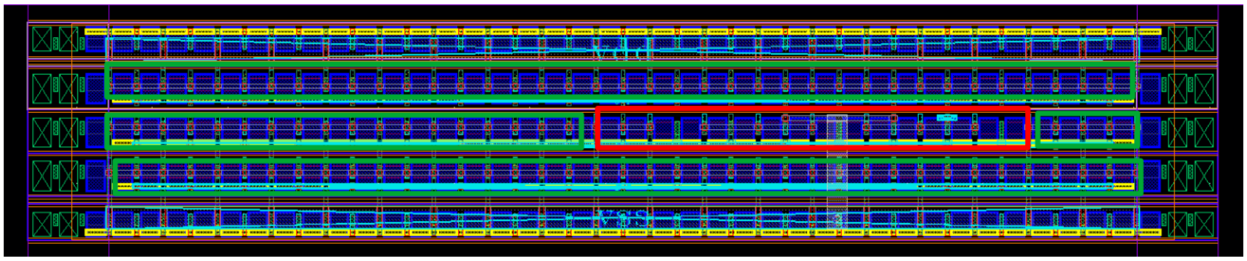


Figure 4.3: Analog Block Layout Snapshot

Differential switch and flip flops

The following is the schematic of the switches and the flip flops: The two latches together make a D-flip flop which samples the input data. The output of the D-flip flop passes through inverters which in turn drive the main switches. The use of D-flip flops is essential to minimize timing delays between the cells and to ensure that the switching happens in unison [5].

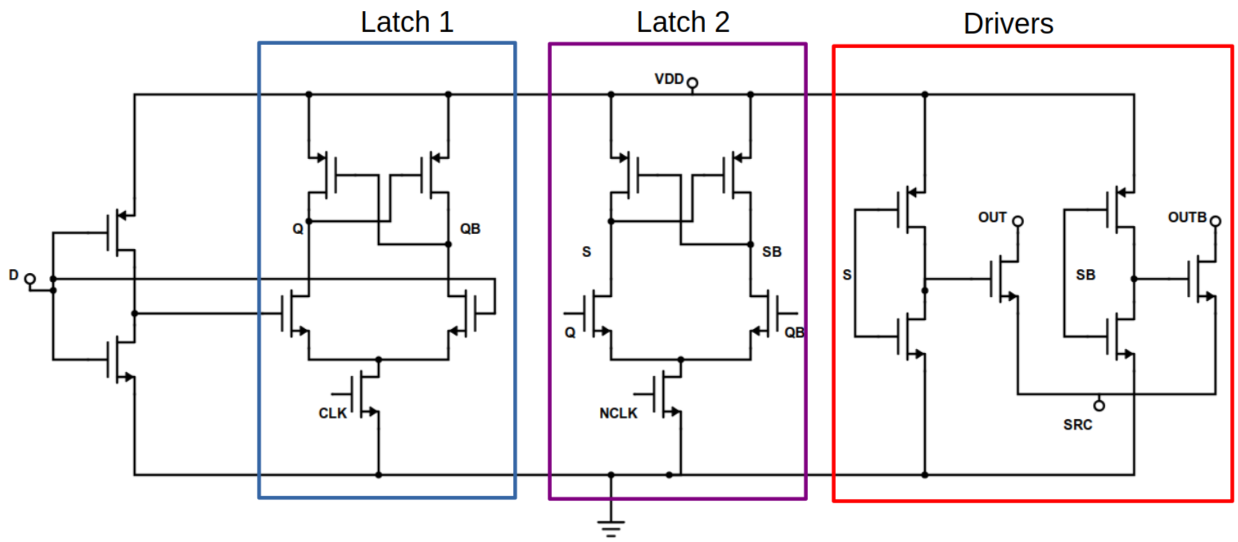


Figure 4.4: Switch drivers and flip flops.

4.2 Dynamic Design with BAG

In order to develop a dynamic design, the code was written to ensure that important design parameters are captured, and the layout can be generated with various parameters. These parameters were chosen such that a unit cell can be designed and ported to various technology nodes with ease:

- **Transistor parameters**: Channel length, width and fingers can be passed as a parameter. In the case of current source, maximum length is chosen, and in the case of the switches, the minimum sized is chosen.

- **Current mirror ratio**: The ratio is also a parameter. The number of fingers of various transistors are dynamically calculated to realize the ratio.
- **Flip flop parameters**: The strength of the switch drivers, the number of fingers of the differential pair input of the latches, and the input buffers are all parameters.
- **Dummy transistors and separation**: The number of dummy transistors and their separation is also a parameter. This can be useful if cell size is a concern, or if more capacitance is needed.

Having all of the above as a parameter is extremely helpful, as it lowers the time it takes between testing the layout and re-designing the layout. For instance, if the current source matching is insufficient after layout, it is extremely easy re-generate the whole design with the resized current sources. Once the Digital and Analog blocks are generated, a unit cell instance is generated by connecting them. Putting both the Analog and Digital blocks as close as possible ensures that the capacitance at the current source drain is minimized. Below is a sample of a unit Cell layout generated by BAG:

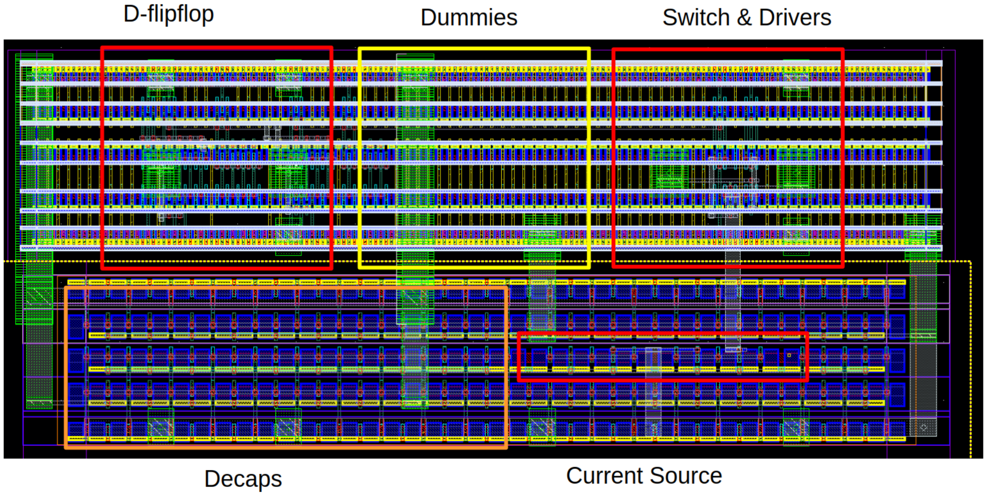


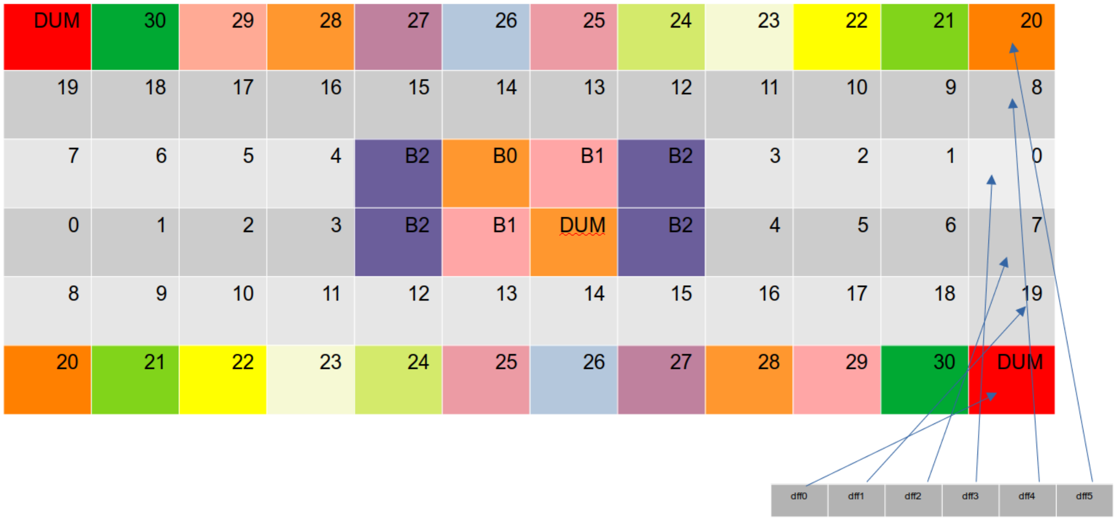
Figure 4.5: Overview of the unit cell.

4.3 Array Design and Routing

Once the unit cell is designed and tested thoroughly in the test bench, the next step is to route the unit cells. This step involves designing the floorplan of the DAC, routing cells and supply connections.

Floorplan

The floor plan is designed to incorporate common centroid structure to minimize the gradients across the layout [6]. Furthermore, special attention is given to ensure that the clock and data are distributed accurately. This has put an upper limit to how large the layout can be. For instance, for a fast clock it becomes increasingly difficult to distribute the clock without encountering clock slew. On the other hand, for designs made for low frequency operation, this is less important. Below is a floor plan sample. There are several points worth mentioning here:



DUM	30	29	28	27	26	25	24	23	22	21	20
19	18	17	16	15	14	13	12	11	10	9	8
7	6	5	4	B2	B0	B1	B2	3	2	1	0
0	1	2	3	B2	B1	DUM	B2	4	5	6	7
8	9	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	DUM

Figure 4.6: Floorplan proposal

- There are 64 thermometer type unit cells in the design. Each thermometer type cell generates 4x LSB current. In order to turn on a thermometer code, 2 of these cells turn on to generate the required 8x LSB current. For instance, in the figure above, the two orange cells numbered 20, receive the same data, and therefore turn on simultaneously.

- There are 7 binary type cells. Each binary type cell generates 1 LSB. 4 of them are connected together to realize the 2nd bit and generate 4x LSB current. (labelled B2 above). Two of these cells are connected together to realize B1. The last cell (B0) realizes 1 LSB.
- Two of the thermometer type cells are dummies (since we only have 30 thermometer codes).
- Each column has a row of D-flip flops. The data from the digital binary to thermometer encoder gets sampled there and then distributes to the columns. This is absolutely crucial since any delay or skew from the previous stage needs to be buffered, otherwise, it will impact the output.
- Each column has a copy of $Clock$ and $\overline{Clock}$, and collects the output currents.
- The supplies (VDD and GND) are connected to form a gird: VDD and GND are connected in the rows. There is also a vertical connector.
- Bias currents are distributed in each column where each cell receives the current. These bias currents are copied from an external reference current.

It is worth mentioning that some of the above decisions in layout design were the result of multiple cycles of simulation and verification. For instance, the decision to add a row of D-flip flops for each column was not initially part of the floorplan.

Below is a snapshot of the resulting layout (note that the current sources are not shown):

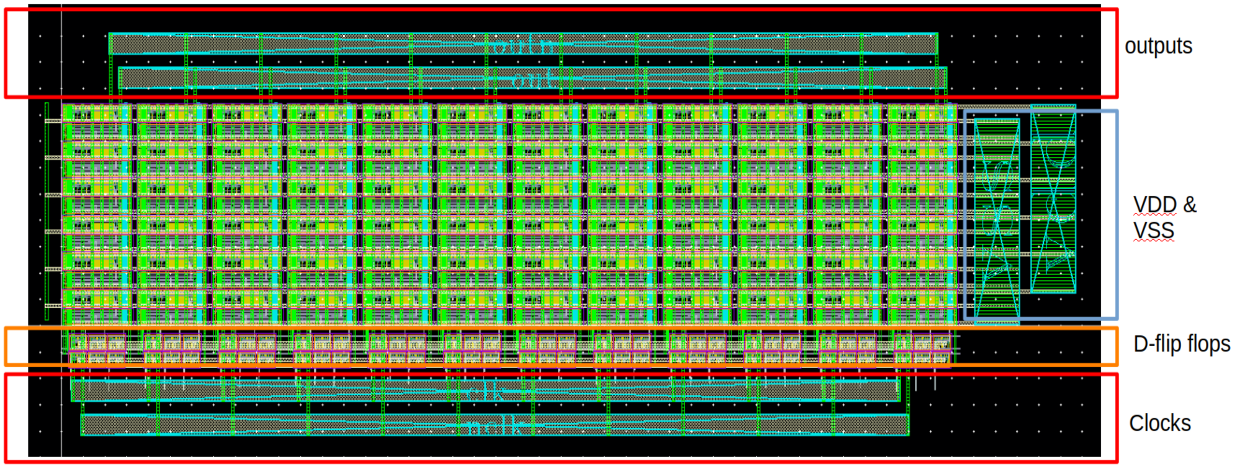


Figure 4.7: DAC layout with routing

Zooming in one cell in the layout, we can see that the digital lines and bias lines are separated by a relatively large distance to avoid coupling between the fast digital lines and the bias current. Furthermore, the differential clocks and output are separated by VDD and VSS to maximize isolation. This technique improves the health of the signal and ensures that coupling between high swing signals is minimized.

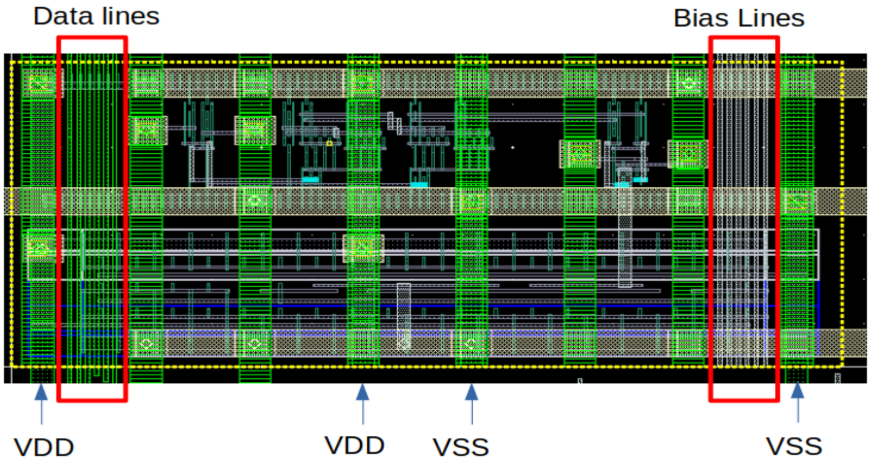


Figure 4.8: Unit cell within the array

4.4 Post Layout Simulations

After layout, a few transient simulation using the test bench from previous chapter has been done over the RC extracted layout. Some of the results are highlighted below:

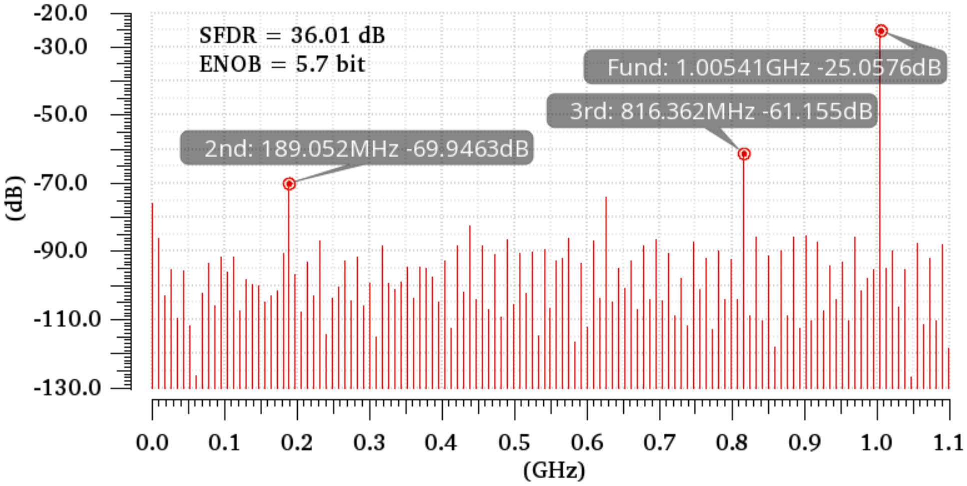


Figure 4.9: 256 DFT of DAC output operating around Nyquist

As we expect, change in input frequency will negatively impact nonlinearity and ENOB, which is why the post layout ENOB is much less than 8.

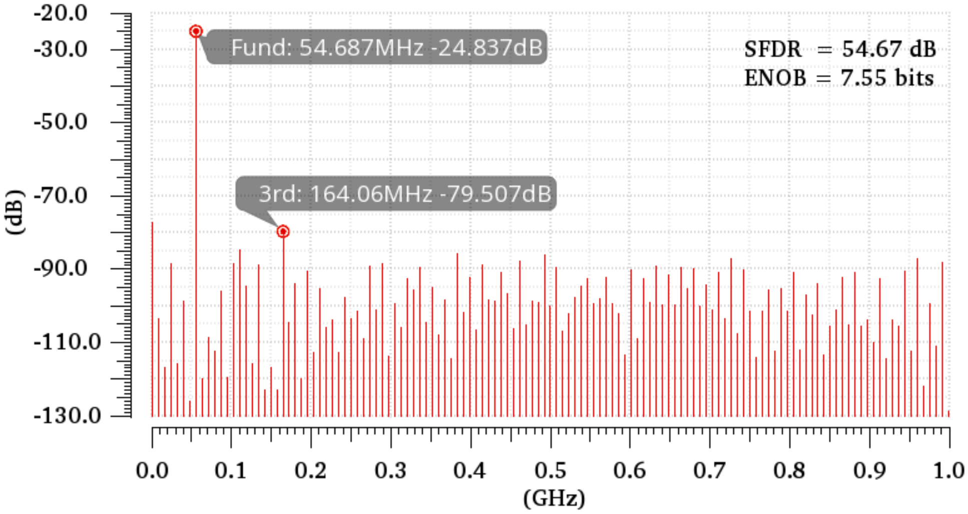


Figure 4.10: DAC output, operating at lower input frequency

Interestingly, increasing the frequency causes the 2nd harmonic to rise and become dominant as opposed to the 3rd order harmonic. It is worth noting that some of the non-linearity present is external to the DAC, since the current mirror PMOS loads are non-linear by nature.

DAC images

Although the current mirror load presents a first order low pass filter for the design, it may not be enough to reject all the images, highlighting the need to have a filter:

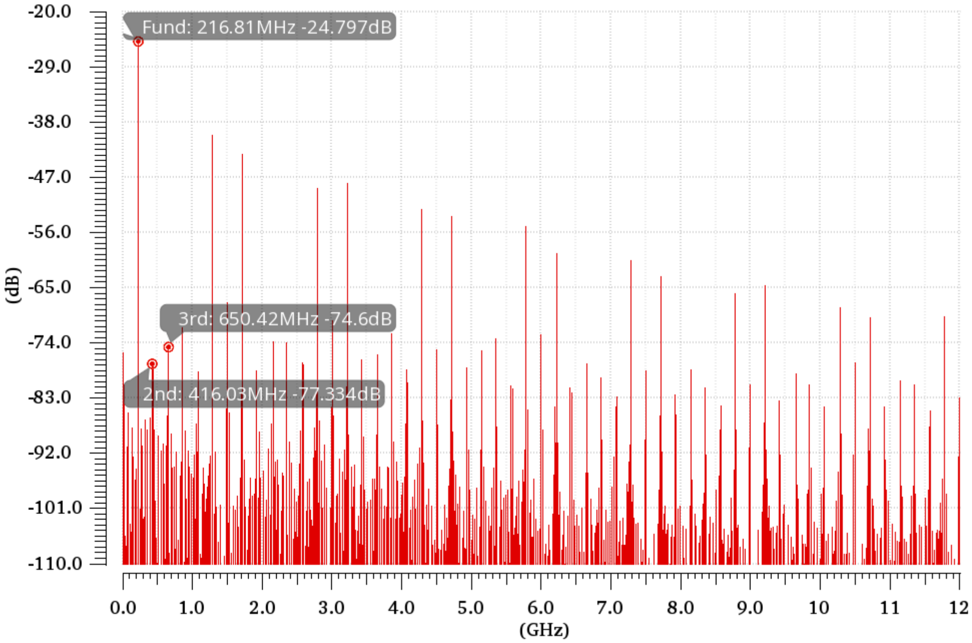


Figure 4.11: over sampled DAC output, showing images

As we see the typical sinc response is still present. For instance, the first DAC image is only 15dB below the fundamental.

Chapter 5

Conclusions

The principles and methodologies for layout and design of a high speed D/A converter is presented here. The main goal here was to develop a reasonable methodology for approaching the design space: a methodology which can be captured via code and reused under different technologies. The developed code can generate the proper schematic and full layout, enabling easy integration with other modules in an extremely complex chip. There are a few points that need to be considered:

- The main obstacle for high speed operation is the timing errors occurring across the design and the parasitic capacitance at critical nodes. By nature, such issues become only visible in post layout simulation. This highlights the need for developing a floorplan strategy.

- Although the principles mentioned here are useful for implementing a general D/A, the implementation and the topology is specific to this application. For instance, the choice to integrate the switches and flip-flops within the unit cell (at minimum distance from the current source) was taken to minimize the drain capacitance at the source, which was needed for high frequency performance. However, for moderate frequencies and higher resolutions, other floorplans might be far more practical. In other words, this approach helps integration and development, but the designer still needs to make critical choices and change the code accordingly.

- The generated layout automatically passes the design rules check. This helps the designer from the time consumption perspective: instead of spending time on minor details, the designer is free to think about the architecture and the layout principles. This also shortens the design cycle from layout to test bench, making the development easier.

5.1 Future Work

There are several steps ahead of this project:

- The design need to be taped out and tested to ensure proper operation. This module will be tested as part of the Hydra tapeout.
- The code itself can be made more dynamic. Throughout the design, certain aspects of the design were assumed to be fixed. For instance, the size of the unit cells were dictated by the power grid designed for the entire chip. The code could essentially treat such aspects as new parameters and design the cell accordingly. Identifying all useful parameters and writing the design based on those to have a truly dynamic unit cell generation can make the reuse incredible simpler and essentially abstract away to complexities for future reuse.

Bibliography

[1] Robert Adams and Khiem Q Nguyen. "A 113-dB SNR oversampling DAC with segmented noise-shaped scrambling". In: *IEEE Journal of Solid-State Circuits* 33.12 (1998), pp. 1871–1878.

[2] Benoit Catteau et al. "An on-line calibration technique for mismatch errors in high-speed DACs". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 55.7 (2008), pp. 1873–1883.

[3] Eric Chang et al. "BAG2: A process-portable framework for generator-based AMS circuit design". In: *2018 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE. 2018, pp. 1–8.

[4] Stanley Yuan-Shih Chen. "Multi-Mode Sub-Nyquist Rate D/A Converter for TV Band Cognitive Radio". In: (2014).

[5] ANDREA FENAROLI. "Techniques for high speed and low power digital to analog converters". PhD thesis. Italy, 2014.

[6] Ray Alan Hastings and Roy Alan Hastings. *The art of analog layout*. Vol. 2. Pearson Prentice Hall New Jersey, 2006.

[7] Seonggeon Kim, Kang-Yoon Lee, and Minjae Lee. "Modeling random clock jitter effect of high-speed current-steering NRZ and RZ DAC". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 65.9 (2018), pp. 2832–2841.

[8] Marko Kosunen et al. *Digital signal processing and digital-to-analog converters for wide-band transmitters*. Helsinki University of Technology, 2006.

[9] Chi-Hung Lin and Klaas Bult. "A 10-b, 500-MSample/s CMOS DAC in 0.6 mm/sup 2". In: *IEEE Journal of Solid-State Circuits* 33.12 (1998), pp. 1948–1958.

[10] Erik Olieman. "Time-interleaved high-speed D/A converters". PhD thesis. Enschede, Mar. 2016.

[11] Marcel JM Pelgrom, Aad CJ Duinmaijer, and Anton PG Welbers. "Matching properties of MOS transistors". In: *IEEE Journal of solid-state circuits* 24.5 (1989), pp. 1433–1439.

[12] J Jacob Wikner and Nianxiong Tan. "Modeling of CMOS digital-to-analog converters for telecommunication". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 46.5 (1999), pp. 489–499.

Appendix A

Appendix A: BAG Code hierarchy & structure

BAG framework itself is accessible from github: https://github.com/ucb-art/BAG_framework. The following graph describes the classes used for the generation of the code:

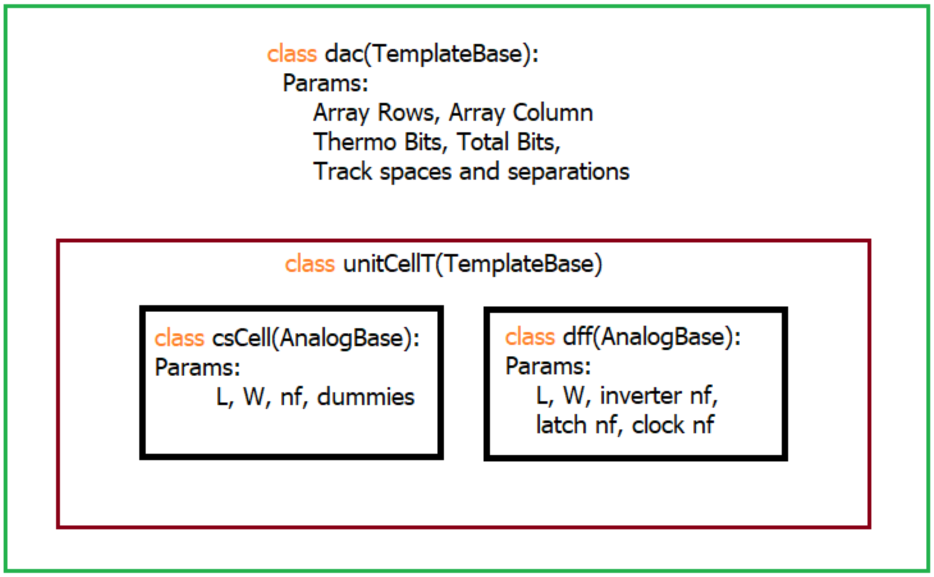


Figure A.1: DAC class hierarchy

This BAG layout design uses various hierarchies to realize a large layout. The most basic building block is called AnalogBase. AnalogBase instances combine to make a TemplateBase. Each AnalogBase has to use the same channel length. There are several points worth mentioning regarding this structure.

1. **csCell** has the largest channel length designed for the current source and the mirror. This block is described in chapter 4 as the "Analog block". This takes in the parameters needed for making a current source including the channel length, width, fingers and the ratio between the mirror and the source and dummy fingers.

2. **dff** is using smallest channel length and is essentially the "Digital Block". It includes the two latches that make a D-flip flop and the minimum sized switches used for current steering. This block takes in the latch parameters (such as the input pair number of fingers, width, clocked transistor fingers), inverter size, and the switch sizes.

3. **unitCellT** This class puts the digital and analog block together and makes the necessary connections and vias to higher layer to be used by the upper level classes.

4. **dac** This class uses the unitCellT class and generates the top-level DAC. It takes in parameters for designing the physical array, such as rows, columns, width of the pins and separation of various tracks.