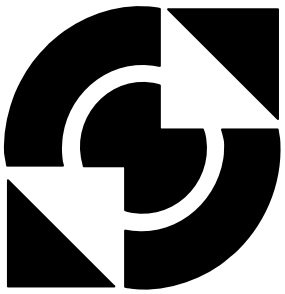
University of Twente

Faculty of Electrical Engineering,
Mathematics & Computer Science



Design of application specific high bandwidth Nyquist DAC

Zhijun Wei
MSc. Thesis
February 2008

Supervisors:
prof. ir. A.J.M. van Tuijl
dr. ing. E.A.M. Klumperink
ir. F. van Houwelingen
Report number: 067.3215
Chair of Integrated Circuit Design
Faculty of Electrical Engineering,
Mathematics & Computer Science
University of Twente
P. O. Box 217
7500 AE Enschede
The Netherlands

ABSTRACT

Digital-to-analog (DAC) converters translate digital codes into different physical quantities in voltage, current or charges. They are used either as a stand alone functional block in a big system or as a subsystem in analog-to-digital (ADC) converters. The DAC in this project is a subsystem in a two-step subranging like ADC. The application specific DAC is also named as internal DAC in this master report, meaning dedicated DACs in ADC systems.

The main goal of this research is to find a high bandwidth internal DAC system that exploits the fact that no SFDR requirement is given and latency is allowed so that a very high signal to noise ratio (SNR) is present at the output during the sampling moment of the ADC. The internal DAC has the following characteristics: low resolution, 2GS/s, and 13-bit accurate reproducibility. Besides that, the optimum output (current, voltage, or charge) needs to be determined when integrating internal DAC itself with subtractor.

Different types of internal DACs, including resistive string, R-2R, capacitive, and current steering types, are examined in terms of reproducibility, speed, power consumption, and noise performance. Universal causes to speed and reproducibility present in all types of DAC are identified: speed limitation is RC settling, and reproducibility limitations are memory effect, signal jitter, digital feedthrough, noise, and switching control signals. Noise is the most dominant factor determining the reproducibility. With the same capacitive load (100fF), comparisons between different DACs are made: current steering DAC has the most output noise, capacitor DAC for largest chip area and highest reproducibility, resistor string DAC for largest power consumption. But the design may come down to one form of output signal (voltage) and one topology (resistive) after taking into account integrating internal DAC with the subtractor. The chosen DAC achieves 2GS/s and 10-bit reproducibility (differential). 13-bit reproducibility can not be achieved due to the required settling speed and input matching requirement.

Contents

Chapter 1 Introduction

1.1 Background

As the silicon process improved, the minimum feature of a metal-oxide-semiconductor field effect transistor (MOSFET) device has kept shrinking over the last two decades, and greatly impacted integrated circuit (IC) design. The latest 45nm technology has entered the production at the end of 2007. This evolution makes ICs become cheaper, faster and smaller, and consume less power, which is good to battery powered devices. The digital integrated circuit benefits more from this down scaling. However, the analog circuit performance does not benefit as much as the digital circuit does due to the reducing supply voltage and noise consideration. The trend is that more analog functional blocks are being implemented in digital domain. Because of the analog nature of signals in the real world, the bridging circuits are required to link the analog world with the digital world. Analog to digital converters (ADC) and digital to analog converters (DAC) are such devices fulfilling this purpose. Therefore, ADCs and DACs are almost indispensable in the nowadays systems.

ADC quantizes the signals from the analog world into digital codes while DAC does the opposite. They are used everywhere in the communication systems. One of major applications for these converters is in ultra-wideband (UWB) system.

UWB is a radio technology for transmitting information spreading over a large bandwidth (>500 MHz) that should, in theory and under the right circumstances, be able to share spectrum with other users. UWB is often used at very low energy levels for short-range high-bandwidth communications. UWB technology thus enables a wide variety of wireless personal area network applications. Examples include:

- Enabling high-speed wireless universal serial bus (WUSB) connectivity for PCs and PC peripherals, including printers, scanners, and external storage devices
- Replacing cables in next-generation Bluetooth Technology devices, such as 3G cell phones, as well as IP/UPnP-based connectivity for the next generation of IP-based PC/CE/mobile devices
- Creating ad-hoc high-bit-rate wireless connectivity for CE, PC, and mobile devices

The high speed ADC is required in UWB system to digitize the ultra-wideband signal. In the project, the signal has the bandwidth of 1GHz and 1V of peak-to-peak swing. The two-step subranging ADC is one of topologies that can be used in the system because it offers not only the speed, but also the accuracy.

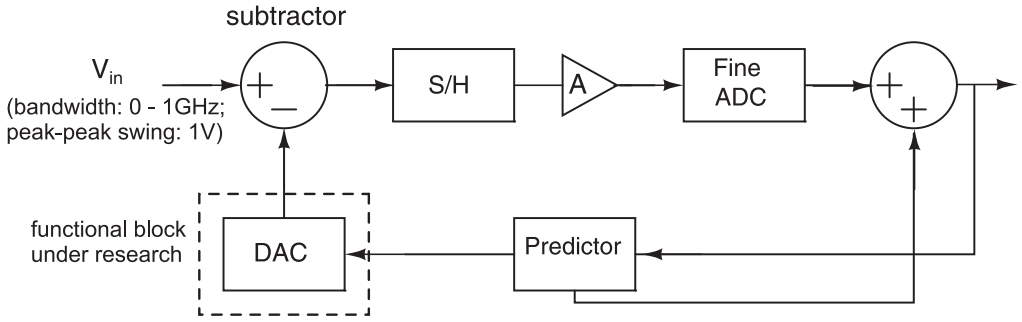


Fig. 1 A two-step subranging like ADC

A normal two step ADC is composed of sample and hold (S/H) circuit, a coarse analog-to-digital converter, a digital-to-analog converter, a subtractor, and a fine ADC. The coarse ADC converts the most significant bits (MSB), and generates the codes to the internal[1] DAC. The residue signal is produced by subtracting the internal DAC values from the input signal and then converted into the least significant bits (LSB). But this conventional ADC may not meet the system requirements [Appendix A-1]. There is a large dynamic and predicable component in the incoming signal to the ADC. Therefore, a new structure is proposed, as shown in Fig. 1. The differences between the conventional structure and the new structure are:

- The S/H circuit is shifted after the subtraction point.
- Instead of a coarse ADC delivering the data for the DAC, the data for the DAC will be created by means of prediction from the past output samples.

The new structure, however, puts demanding requirements on the internal DAC. Its sampling speed needs to be at least two times the highest frequency of the incoming signal and the output level reproducibility[2] better than the overall resolution of the ADC. Its resolution about 3 – 4 bits is sufficient in this application. So the implementation of the fast internal DAC with reproducible output levels is important to realize the architecture in Fig. 1. This master project focuses on the possible DACs that meet these requirements.

1.2 Digital-to-analog converter

The internal DAC in the system obtains the input code from the predictor and provides the corresponding output level to the subtractor. It is required to run at 2GS/s with the reproducibility as high as possible. The design considerations for high speed internal DACs are different from the stand alone DACs. Among them, spurious free dynamic range (SFDR) is one major difference. Normally, stand alone high speed DAC should preferably not generate any spurious component in their output spectrum. Spurious components are unwanted in UWB system because these tones

[1] The term, 'internal', in the thesis is used to refer to those DACs utilized as a subblock in ADCs, as opposed to 'stand alone', which means an independent functional block on a higher level.
[2] Reproducibility is referred in the thesis as DAC's ability to produce the same output level for an input code.

can not be filtered out efficiently and act as interferers. Subsequently, the DAC should not have any switching glitches present at the output in order to have good SFDR. But SFDR for internal DACs is not as important as for stand alone DACs. As seen in Fig. 1, the output of the internal DAC gets sampled. The sample rate of the S/H circuit and the internal DAC are equal, but the sample moments are not the same. This means that one sample of the internal DAC can be visualized as following.

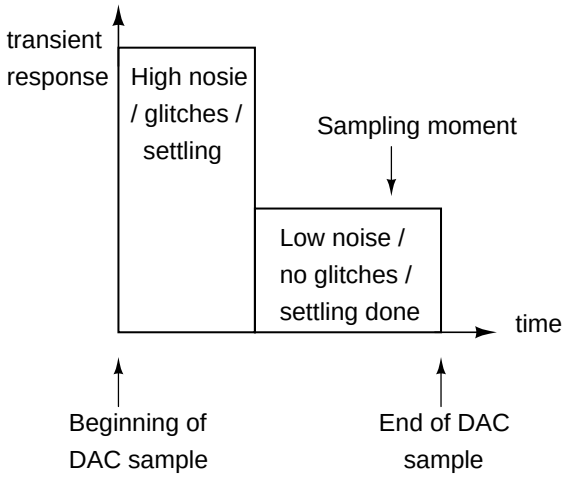


Fig. 2 Transient response

The internal DAC is required to provide an accurate output voltage only at the moments when the S/H circuit takes samples. Outside of the sample moments, noise, glitches and settling due to limited bandwidth are allowed. It can be generally said that any internal DAC for the architecture in Fig. 1 does not have a SFDR requirement. As a result, a high bandwidth DAC in the project is required to have the following characteristics.

- No SFDR requirement
- Stable or reproducible output levels so that the signal-to-noise ratio (SNR) at the S/H moments is as high as possible (target = order of 13-bit accuracy, 122μV for 1LSB with 1V input swing)
- 3 to 4 bits quantization levels
- 2GHz sampling frequency

1.3 Research objectives

The primary research is on a high bandwidth internal DAC with 3 – 4 bit resolution and 13-bit reproducibility, exploiting the facts that no SFDR requirement is given, the latency is allowed, and the output offset due to component mismatch can be calibrated. The chosen technology is PHILIPS CMOS 65nm standard low voltage process. The subtraction is briefly discussed as well since the choice of internal DAC for the project depends on how it can interface with the subtractor and still achieves the performance after the subtractor. The research intends:

- To identify the differences between stand alone and internal DACs.
- To find out the common speed and reproducibility limiting factors present in different internal DACs.
- To examine the speed, reproducibility, and power consumption of different internal DACs.
- To determine what type of DAC output signal is optimum to interface with the subtractor.

1.4 Outline of the thesis

The report has been divided into five chapters. Chapter 2 introduces the definition of errors and specifications for stand alone DACs, the commonly used digital codes and the causes of power consumption in DACs. Besides these introductions, the Nyquist-rate DACs, including resistive, capactive and current steering type, are discussed. The comparison between different stand alone DACs is given in terms of precision, speed, and power consumption at last.

Chapter 3 focuses on the internal DAC in the system. First, the internal DAC requirements are derived. After that, the differences between stand alone and internal DACs are identified, the universal speed and reproducibility limiting mechanisms are presented. Later, different internal DAC architectures are studied, designed and simulated with the same boundary conditions. The comparisons are made at last.

Chapter 4 explains how the subtraction can be done and which type of internal DAC is a good choice for integration with analog subtraction. The proposed internal DAC is simulated and achieves 10-bit reproducibility.

Finally, conclusion and recommendations are presented in chapter 5.

Chapter 2 Introduction to Digital to Analog Converter

DACs are devices for converting a digital code to an analog signal (current, voltage or electric charge). Ideally, the output quantity is a linear function of the input codes. These codes are usually written to the DAC with a clock signal, and the output is in a discrete form so the system is discrete in both time and amplitude. The internal DACs share a lot of similarities with the stand alone DACs in terms of the architecture and some performance metrics. Understanding the stand alone DAC is helpful to gain the insights into the internal DAC design so this chapter will focus on introducing the fundamentals of the stand alone DACs.

Before introducing the various DAC architectures, it is necessary to consider the standard measures to characterize DAC performances. The first section is contributed to the definition of errors and specifications, which are useful to understand the strengths and weaknesses of different DAC architectures. The next sections address the commonly used digital codes and the causes of power consumption in DACs. Then Nyquist-rate DACs, including resistive, capactive and current steering type, are briefly discussed since the assignment is to investigate the internal DACs for a Nyquist-rate ADC. Finally, the comparison between different DACs is given in terms of precision, speed, and power consumption.

2.1 DAC performances

Generally, DAC performances due to circuit nonidealities can be categorized into two different types, static and dynamic. The static performances are signal-independent (non-memory effects), and the dynamic performances are signal-dependent (memory effects). A typical static error is manifested as the deviation from the ideal DC transfer curve. The examples are gain error, offset, differential (DNL) and integral nonlinearity (INL). These errors are dominant at low frequencies, while the dynamic errors come into effect as the signal and clock frequency increase. The dynamic errors include clock feedthrough (CFT), glitches, settling errors, etc. The static performance can be regarded as the best-case scenario of a data converter.

2.1.1 Static Performances

In this section the most common static performances are outlined, such as the quantization error, gain and offset error, DNL, and INL.

- The quantization error is the difference between the discrete DAC output levels and the continuous ramping signal, as shown in Fig. 2.1.

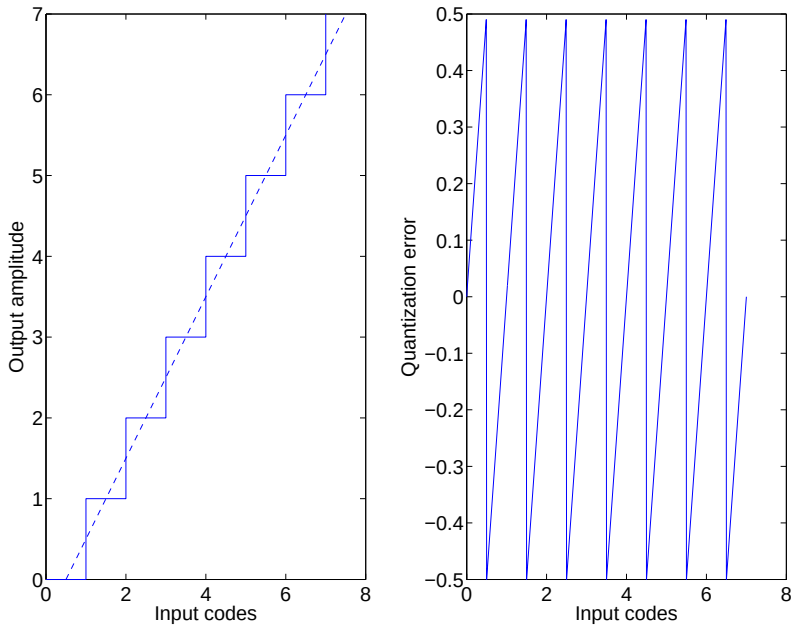


Fig. 2.1 DAC transfer curve and its quantization error

The signal to noise ratio (SNR) of an ideal DAC is limited by its quantization error. SNR can be evaluated by assuming quantization noise of uniform distribution in [– LSB/2, +LSB/2] and sinusoidal input signal [1].

$$SNR = 6.02N + 1.76 \text{ dB} \quad \text{(e2.1)}$$

- The offset error is the difference between the ideal LSB voltage to the actual LSB voltage.
- The gain error indicates how well the slope of an actual transfer function matches the slope of the ideal transfer function (usually unity).

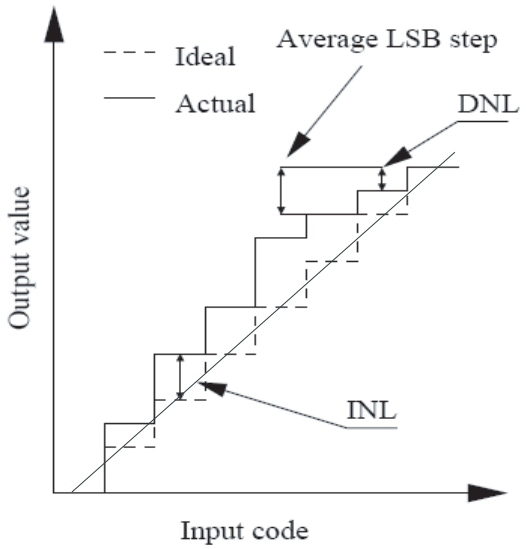


Fig. 2.2 DNL and INL

- Differential nonlinearity (DNL) of a DAC is the maximum deviation in the output step size from the ideal or the average value of one least significant bit (LSB).

$$DNL = \max\left(abs\left(DNL_n\right)\right) = \max\left(abs\left(\frac{V_{real,n+1} - V_{real,n}}{V_{LSB}} - 1 \right) \right), \quad \text{(e2.2)}$$

n = 1, 2, … 2^N-1

where $V_{real,n}$: the actual DAC level at the n^{th} code, V_{LSB}: the ideal or average LSB step, N: the number of bits. DNL_n shows the step difference from the average step at the code transition from *n* to *n*+1. If the value of DNL_n is less than −1, the converter is no longer monotonic. DNL is caused by the component mismatch and the nonlinearities in the circuit.

- Integral nonlinearity (INL) of a DAC is defined as the maximum deviation between the analog output value and the straight line drawn between output values corresponding to the smallest and the largest input code. It causes harmonic distortion in DACs.

$$INL = \max\left(abs\left(INL_n\right)\right) = \max\left(abs\left(\frac{V_{real,n} - V_{ideal,n}}{V_{LSB}} - 1 \right) \right), \quad \text{(e2.3)}$$

n = 1, 2, … 2^N-1

2.1.2 Dynamic Performances

DNL and INL define the limit of the converter linearity at low frequencies. It is further degraded by the dynamic nonidealities as the signal frequency and sampling rate increases.

- Settling error is the deviation from the final value at the end of the settling time. Settling time is the interval between a command to update its output value and the instant it reaches its final value, within a specified percentage.
- Glitches occur during the code transitions. They show overshoot, undershoot, or both. DAC with binary codes experiences worse glitches than the one with thermometer codes.
- The clock/digital feedthrough can be illustrated by looking at the simple sampling circuit composed of a NMOS switch and a holding capacitor. Due to the capacitive coupling through the gate-drain/gate-source overlap capacitance, the clock affects the analog output signal during the transitions.

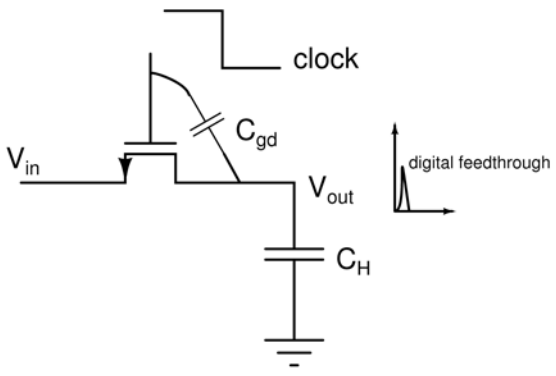


Fig. 2.3 clock feedthrough in the sampling circuit

- A signal-to-noise ratio (SNR) is the ratio of the signal power (usually the power of a sinusoidal signal) to the noise power integrated over half of the sampling frequency, f_s, in decibels.
- The total harmonic distortion (THD) is the ratio of the signal power to the total power of all the harmonic components and measured in decibels.
- The spurious free dynamic range (SFDR) is the level of highest spurious tone relative to the signal power and measured in decibels. The spurious tones are most often due to the nonlinearity in the converter.
- The signal-to-noise and distortion ratio (SINAD) is the sum of noise and distortion power relative to the signal power.
- The effective number of bits (ENOB) specifies the dynamic performance of a DAC at a specific input frequency and sampling rate. ENOB is evaluated by

$$ENOB = \frac{SINAD - 1.76}{6.02} \quad \text{(e2.4)}$$

2.2 Common digital codes

The digital input to a DAC can be any format but must eventually be of a form easily convertible to analog. Below shows some commonly used digital codes in DACs.

- Two's complement code: the weights of two's complement codes are given by
  $W_m = 2^{m-1}$, m = 1, 2, …, N – 1 and $W_N = -2^{N-1}$ (e2.5)
  Both negative and positive numbers can be converted with the two's complement-coded DACs.
- Offset binary code: the weights of offset binary codes are given by
  $W_m = 2^{m-1}$, m = 1, 2, …, N (e2.6)
  Only positive numbers can be realized with the offset binary coded DAC. For example, it is used in the current steering DACs.
- Thermometer code: all weights are equally large and given by
  $W_m = 1$, m = 1, 2, …, M, where $M = 2^N - 1$ (e2.7)
  The thermometer code is widely used in DACs since a number of weights are equal and hence components can be laid out with special techniques to achieve good matching. With thermometer codes, glitches can be minimized.
- Walking one code: the weights of walking one codes are given by
  $W_m = m$, m = 1, 2, …, M, where $M = 2^N - 1$ (e2.8)
  For each word, only one bit is '1' and the rest are '0's. It is mostly used for weight selection. The resistor-string DAC uses the code.

2.3 Power dissipation

DAC bridges digital and analog signal in a system so it contains both analog and digital circuitry. In digital circuit, there are three major mechanisms of power consumption. The first one is related to the switching activity, charging and discharging the parasitic capacitors.

$$P_{switch} = \alpha \cdot C_L \cdot f_{clk} \cdot V_{DD}^2 \quad \text{(e2.9)}$$

where α: the switching activity, C_L: the average load capacitance, f_{clk}: the clock frequency, V_{DD}: the supply voltage. Another one results from short circuit current.

$$P_{sc} = I_{sc} \cdot V_{DD} \quad \text{(e2.10)}$$

The short circuit current I_{sc} is due to the direct-path from the supply to the ground, which appears when both NMOS and PMOS transistors are active at the same time. The last mechanism is the leakage current in the circuit.

$$P_{leakage} = I_{leakage} \cdot V_{DD} \quad \text{(e2.11)}$$

$I_{leakage}$ is determined by the fabrication technology. It can arise from the substrate injection and the leakage current in the subthreshold region. In analog circuits, there are two major causes of power consumption. One is the DC power dissipation drawn from the power supply.

$$P_{DC} = I_D \cdot V_{DD} \quad \text{(e2.12)}$$

The other is related to charging/discharging the capacitors in the circuit, which is similar to switching power in digital circuit. This power consumption becomes pronounced in switched capacitor circuits.

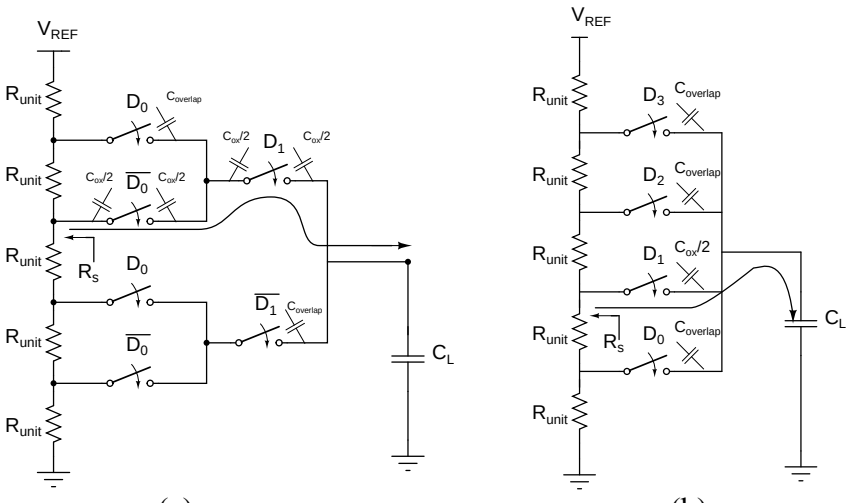
$$P_C = C \cdot f_{clk} \cdot V_{DD}^2 \quad \text{(e2.13)}$$

2.4 Nyquist-rate DACs

Digital-to-analog converters are built in several different topologies. They are resistive, capacitive, or current steering type. Each topology has its own strengths and weaknesses, which can be summarized in several aspects: INL, DNL, matching requirements versus the achievable precision, monotonicity, speed and power consumption. This section briefly introduces the different architectures and discusses their pros and cons.

2.4.1 Resistor string DAC

This is one of common seen DACs, and obviously resistive type. An N-bit version of this DAC consists of 2^N equal unit resistors in series. One of these voltage taps is selected by a decoder network to connect appropriate voltage tap to the output.



(a) (b)

Fig. 2.4 Resistor string DAC with (a) binary decoding; (b) walking one decoding

The switching network can be controlled by the binary or walking one codes, as shown in Fig. 2.4. The output buffer is required to drive large capacitive or resistive load.

The monotonicity is guaranteed in this DAC. The accuracy of the DAC depends on the resistor matching. The output voltages are the most accurate at the ends and the worst in the middle of the string [7]. To ensure INL less than 0.5LSB with 99.7% probability (3σ), the following condition needs to be met [Appendix B-1].

$$\frac{\sigma_R}{R} \le \frac{1}{3\sqrt{2^{N+1}}} \qquad \text{(e2.14)}$$

The corresponding DNL with 99.7% (3σ) probability is [Appendix B-1]

$$DNL = 3\frac{\sigma_R}{R} \qquad \text{(e2.15)}$$

The following table is generated to show the required resistor matching for different resolutions to achieve 0.5LSB INL and the corresponding DNL. From the calculation, 1% resistor matching achieves 9-bit resolution.

Table 2.1 Resolution vs. resistor matching

Resolution (bits)	required matching (%)	DNL (LSB)
8	1.47	0.044
10	0.74	0.022
12	0.37	0.011

The data conversion speed of the resistor string DAC is determined by the output settling speed. Fig. 2.4 shows a conduction path for both decoding networks. There are N switches along the conduction path for the DAC with binary decoding. Assuming that all the switch sizes are equal, its output settling time is expressed as

$$\tau_{bin} = R_{L\_total}C_{L\_total} = (R_s + NR_{sw})(C_L + C_{ox}/2 + C_{overlap}) \qquad \text{(e2.16)}$$

where R_{sw}: the switch on resistance, R_s: the equivalent resistance looking into the resistor string, C_L: the load capacitance, C_{ox}: the gate capacitance of the switches, $C_{overlap}$: the gate-drain/gate-source overlap capacitance. For the DAC with walking one decoding, there is only one switch along the path. Although its total resistance at the output is smaller, the total capacitance at the output is larger because all the switches are connected to the output node. The settling time is expressed as:

$$\tau_{walk} = R_{L\_total}C_{L\_total} \qquad \text{(e2.17)}$$

$$R_{L\_total} = R_s + R_{sw},\ C_{L\_total} = C_L + C_{ox}/2 + (2^N - 1)C_{overlap}$$

where R_{sw}, R_s, C_L, C_{ox}, and $C_{overlap}$ are the same definitions as in eqn. 2.17. For both architectures, the largest settling time is caused by the conduction path from the middle tap to the output because R_s reaches maximum. Reducing R_{unit} and increasing switch ratio helps increase the conversion speed. The difference between the settling times is: R_{L_total} for DAC with binary decoding increases proportionally with the resolution, while C_{L_total} for DAC with walking one decoding grows exponentially. The following simulation data plots the time constant ratio between two architectures with the same DC power consumption. The ratio (τ_{walk}/τ_{bin}) indicates the speed difference.

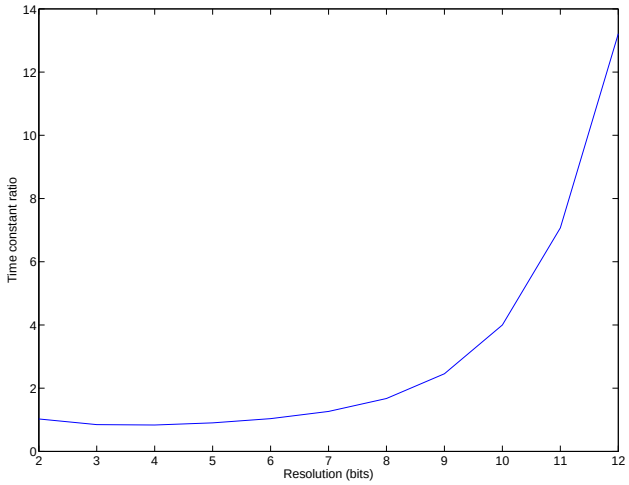


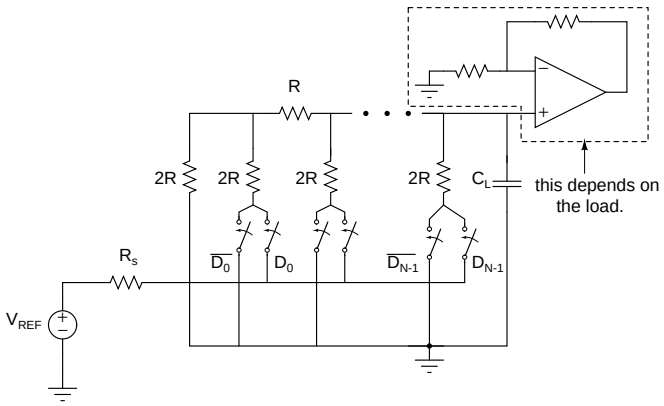
Fig. 2.5 Time constant ratio (τ_{walk}/τ_{bin})

The simulation shows that DAC with walking one decoding is preferred for low resolution design because its decoding network is simpler, the number of switches is much less, and the time constant is a little smaller for N between 2 and 6 as compared to the binary decoding; and DAC with binary decoding is preferred for high resolution design because its time constant is more than ten times less. If the load includes resistor, a voltage buffer is required at the DAC ouput. Then, the speed is limited by an OPAMP, usually less than a few hundred Mega Hertz.

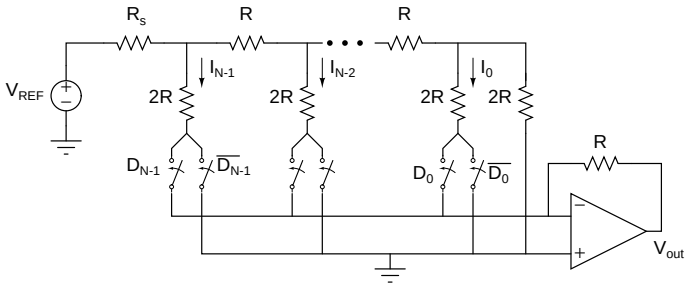
The total power consumption consists of DC power and dynamic power to charge/discharge the load and parasitic capacitors. DC power is inversely proportional to the total string resistance. The dynamic power is proportional to the capacitance, the clock frequency, and the voltage amplitude they are charged to. If buffer is needed to drive the load, OPAMP consumes fairly amount of power as well.

2.4.2 R-2R ladder DAC

This DAC only consists of two resistor values, R and 2R. Digital bits control the switches to connect one end of 2R resistor to either the reference voltage/one OPAMP input or the ground in Fig. 2.6. The voltages or currents of different weights are summed up in the form of voltage or current.



(a)



(b)

Fig. 2.6 Resistor R-2R ladder network (a) voltage mode; (b) current mode

The monotonicity is not guaranteed in this DAC due to the binary codes. The worst case DNL condition tends to occur at midscale when the code transitions from 01…11 to 10…00, and the worst case INL tends to occur when the input code is 01…11 [1, 2]. It is formidable to derive DNL/INL with regard to resistor mismatch due to lengthy iterative differentiation process so MATLAB program [Appendix B-3] is developed to collect the data on resolution versus the resistor matching (500 runs). In the program, the same relative matching with a Gaussian distribution is assigned for R and 2R resistors, and the corresponding DNL and INL are computed. Table 2.2 shows the required resistor matching for different resolutions to achieve 0.5LSB INL and their DNL. From the simulation, 0.08% resistor matching achieves 9-bit resolution.

Table 2.2 Resolution vs. resistor matching

Resolution (bits)	required matching (%)	DNL (LSB)
8	0.2	0.73
10	0.05	0.72
12	0.01	0.8

Note: the data are for voltage mode R-2R ladder DAC. The similar program can be developed for current mode R-2R ladder DAC. It is expected that the data could be close to this table because of the binary coding and similar architecture.

If driving the capacitive load, the conversion speed of voltage mode R-2R ladder DAC is determined by the output settling speed. Since the output resistance of the resistive network is equal to R, the time constant is RC_L. If the load includes resistor, the buffer is necessary. Current mode R-2R ladder DAC requires virtual ground to set the currents of different weight so OPAMP is indispensable. The speed for both cases is limited by OPAMP to about a few hundred mega Hertz.

The total power consumption consists of DC power and dynamic power to charge/discharge the load and parasitic capacitors. DC power is inversely proportional to the resistance. The dynamic power is proportional to the capacitance, the clock frequency, and the voltage amplitude they are charged to. If buffer is needed for driving the load, OPAMP consumes fairly amount of power as well.

2.4.3 Capacitive DAC

Capacitive DACs are favored in some applications due to its low power consumption and good capacitor matching. Digital bits control which capacitor to be charged during the reset phase and the stored charges are shared with the capacitor array during the charge sharing phase resulting in different output levels.

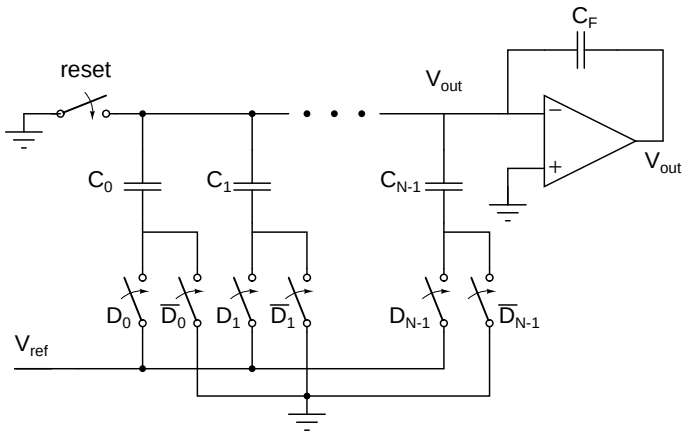


Fig. 2.7 Capactive DAC with unitary elements

The elementary components are capacitors in either unitary or binary weight. The problem with capacitive DACs is that leakage causes it to lose its accuracy within a few milliseconds of being set [3]. In Fig. 2.7, the DAC output is connected to the ground via a reset switch and to the MOSFET gate of the buffer input. It is known that the reverse biased drain/source-to-bulk pn junction leaks in the switch. In addition, submicron MOSFET has the gate leakage. This becomes even pronounced for deep submicron technology. The effects may make capacitive DACs unsuitable for general-purpose DAC applications, but it is not a problem for it to be used in certain ADCs, where the conversion is complete in a few micro seconds or less. That is short before leakage has any appreciable effect.

The monotonicity is guaranteed for DAC with unitary elements, but not for binary elements. For capacitive DAC with thermometer code, the relationship between the linearity and the component matching is similar to the analysis for resistor string DAC. The worst INL happens in the middle of output voltage. The worst DNL occurs for the worst matching between neighboring capacitors. For capacitive DAC with binary code, it is similar to R-2R ladder DAC. The worst case DNL condition tends to occur at the midscale when the code transitions from 01…11 to 10…00, and the worst case INL tends to occur when the input code is 01…11. Eqn. 2.15 and Eqn. 2.16 apply to capacitive DAC with unitary elements. Eqn. 2.15 also applies to capacitive DAC with binary weights. However, its DNL becomes [Appendix B-2]:

$$DNL = 2^{N/2} \cdot 3\frac{\sigma_C}{C} \tag{2.18}$$

Table 2.3 Resolution vs. capacitor matching

Resolution (bits)	Required matching: unitary/binary (%)	DNL: unitary/binary (LSB)
8	1.47/1.47	0.044/0.7
10	0.74/0.74	0.022/0.71
12	0.37/0.37	0.011/0.71

Note: assume that the different sizes of capacitance have the same percentage error.

The charging/discharging the capacitor array takes much less time than the OPAMP settling time as long as the switch sizes are made sufficiently large. Therefore, OPAMP is the speed limiting factor in capacitive DACs. Its power consumption

consists of DC power and dynamic power to charge the capacitor array. The biasing current in the output buffer accounts for DC power. The dynamic power is proportional to the capacitance, the clock frequency, and the voltage amplitude they are charged to.

2.4.4 Current steering DAC

A current steering topology employs a set of currents that are switched by digital input codes to either the load or the ground. The currents are never turned off to save set-up time. Besides this, the resistive load makes the current steering DAC suitable for high speed applications. The currents are in either unitary or binary weight. The advantages of these two different weights apply to current steering DAC too. Thermometer code shows smaller glitches at major transitions, while binary code reduces chip area and has less complexity. For high resolution DAC, a hybrid code or segmented architecture, i.e. binary code for LSBs and thermometer code for MSBs, is used to improve DNL, and reduce the glitch area and the chip area.

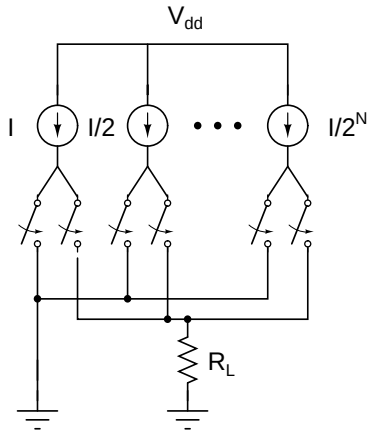


Fig. 2.8 Current steering DAC with binary weights

The monotonicity is guaranteed for DAC with unitary elements, but not for binary elements. For capacitive and current steering DAC, the output quantity (voltage/current) bears the same relationship to the components (capacitors/currents) so the resolution versus the required current matching for unitary and binary weights has the same table as capacitive DAC.

Table 2.4 Resolution vs. current matching

Resolution (bits)	Required matching: unitary/binary (%)	DNL: unitary/binary (LSB)
8	1.47/1.47	0.044/0.7
10	0.74/0.74	0.022/0.71
12	0.37/0.37	0.011/0.71

The difference is: capacitor mismatch dominantly manifests as plate area mismatch, while current mismatch in MOSFET fundamentally results from threshold voltage mismatch and current mismatch. Another limiting factor is the finite output impedance of current source. The relation between the output resistance and the achievable INL is [4]

$$INL = \frac{I_{unit} R_L^2 N^2}{4R_{imp}} \tag{2.19}$$

where R_L: the load resistor, I_{unit}: the LSB current, and N: the total number of unit current sources. The cascode configuration of the switch and current source can meet the INL specification in most cases. However, this is true only over a limited frequency bandwidth, which is determined by the pole at the drain of the current source. The power dissipation of analog circuit can be estimated by the product of V_{DD} and the total current.

2.5 Summary

The static and dynamic performance metrics of stand alone DAC and common digital codes are introduced in this chapter. Also, the causes of power consumptions in DACs are presented. One results from digital circuit, and the other from analog circuit.

After that, several Nyquist-rate DACs are discussed. They are resistive, capacitive, and current types. Both resistor string DAC and R-2R ladder DAC are resistive type. The major difference is that resistor string DAC uses thermometer code, the other binary code. Capacitive and current types can be in unitary and binary weight. These Nyquist-rate DACs are compared in respects of the achievable resolution with different component matching, speed limitation and power consumption. These are summarized in the following table.

Table 2.5 Comparisons between different DACs

Architecture	**resolution**	**Speed**	**Power consumption**
Resistor string	medium	limited by RC_L to drive capacitive load; limited by OPAMP to drive resistive load	$\frac{V_{DD}^2}{R_{tot}}$
R-2R	medium	limited by OPAMP	$\frac{V_{DD}^2}{R_{eq}}$, R_{eq}: equivalent resistance looking into resistor network from the supply.
Capacitive	high	limited by OPAMP	low, no DC power
Current steering	high	high with resistive load	$I_{tot}V_{DD}$

Note: Power consumption only shows the DAC core contribution without the consideration of output buffer.

The calculation in the chapter shows: to ensure 0.5INL, resistor string DAC achieves 9-bit resolution and R-2R ladder DAC achieves 6-bit resolution for 1% resistor mismatch; capacitive DAC and current steering DAC achieve 16-bit resolution for 0.1% capacitor or current mismatch. In addition to the component mismatch, there is one more factor influencing INL in current steering DAC, which is the finite output impedance of the current source.

Chapter 3 Comparison of Internal Digital-to-Analog Converters

This chapter first gives an introduction to the two-step subranging like ADC where the internal DAC is designed, and the internal DAC requirements are derived. After that, the differences between stand alone and internal DACs are identified. There are common limiting mechanisms for speed and reproducibility, which are present in different types of internal DACs. They are summarized in section 3.3 and section 3.4. Section 3.5 includes the design considerations, such as the choices of switch, the load to use in the design for general discussion, and how to measure the output reproducibility. Later, different architectures are studied, designed and simulated. The summary is given at last.

3.1 System overview and internal DAC requirements

This section introduces the ADC system where the internal DAC is designed, and derives the internal DAC requirements.

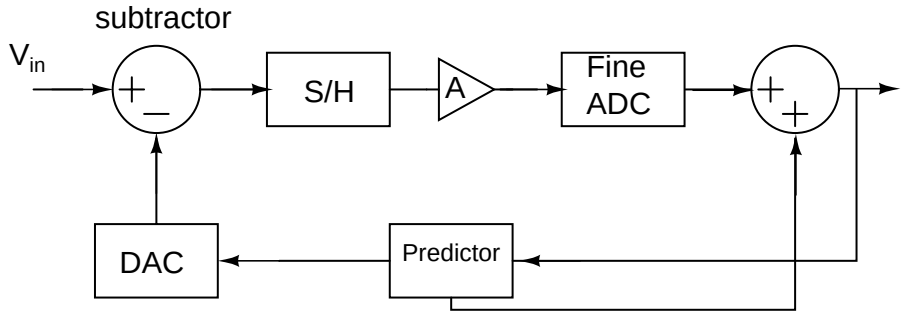


Fig. 3.1 Two-step subranging like ADC

The two-step subranging like ADC is based on two separate conversions – a coarse prediction for MSBs and a fine conversion of the residue signal for LSBs. The input signal is first subtracted by an internal DAC value, and then the residue signal is sampled and amplified to the range of the fine ADC. The gain stage relaxes the accuracy requirement for the fine ADC. The digital input to the internal DAC comes from the predictor. The predictor is a digital estimation system employing certain algorithm to predict the input signal. It is assumed in the project that the predictor always has right estimations for MSBs.

The updating rate of internal DAC is determined by the sampling theorem. The rate is 2GS/s, which is two times of the maximum incoming signal frequency. And the reproducibility of DAC output levels is demanded to be on the level of 13-bit resolution [Appendix C-1]. In other words, the variation of each DAC output level at the sampling moments should not be larger than 122μV for single ended structure or 61μV for differential structure. Due to component mismatch, the DAC output levels

deviate from the ideal levels. But this fixed offset can be calibrated in the system. These are illustrated by the following graphs.

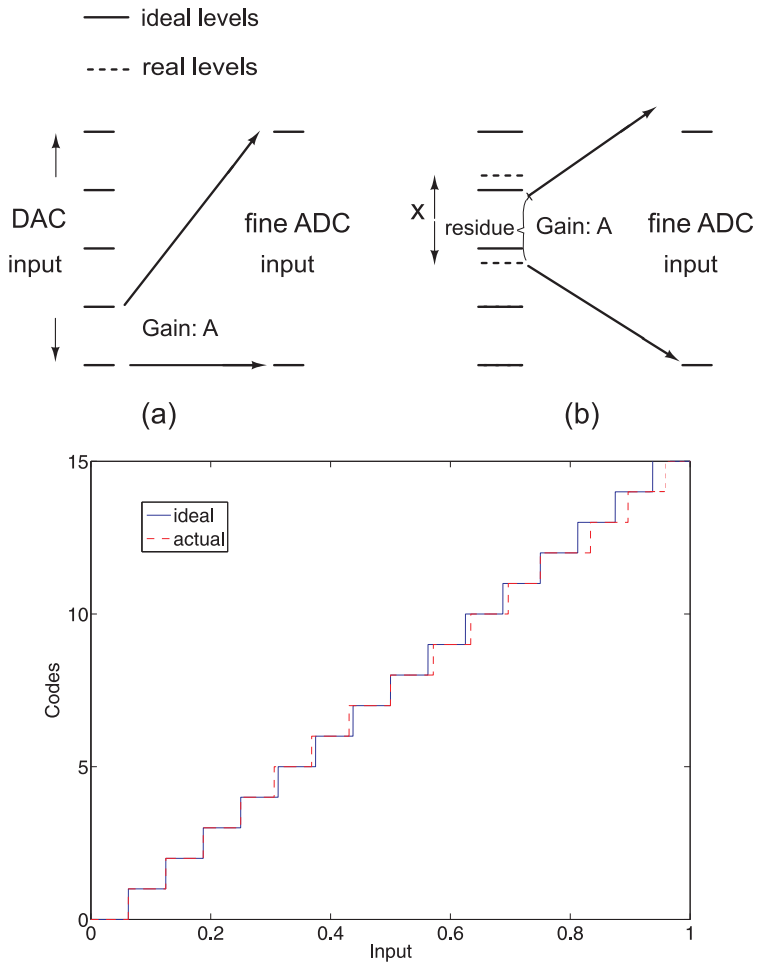


Fig. 3.2 The effect of DAC output offset on the system. (a) ideal output levels; (b) real output levels; (c) ADC output transfer curve (2 bits for both stages)

Fig. 3.2 (a) depicts the ideal conversion. The input signal is predicted in the right coarse subrange, the internal DAC generates the precise output levels, and the amplified residue signal exactly fits into the input range of the fine ADC. However, there are imperfections in the DAC like component mismatch. As a consequence, the actual DAC outputs deviate from their ideal levels, and the amplified residue signal may exceed the input range of the fine ADC. An example is given in Fig. 3.2 (b). If the input signal falls in the subrange X and is close to its up limit, the amplified residue signal will be larger than the input range of the fine ADC. This results in ADC DNL and INL errors, as shown in Fig. 3.2 (c). The Matlab behavior model codes is attached in Appendix C-2. To calibrate the error resulting from the fixed DAC output offsets, the overrange must be accommodated in the fine ADC. The solution is to reduce the amplifier gain accordingly. There is a relationship between the maximum DAC output offset and the optimum gain. Intuitively, larger the offset is, smaller gain is used to accommodate the overrange in the fine ADC. Therefore, the component mismatch will not be regarded as a design bottleneck because of the overrange calibration technique.

3.2 Differences between internal and stand-alone DAC

Most internal DACs are derived from the stand alone DACs. They share the similar architectures. But there are differences due to different applications.

First, a stand alone DAC has 2^N precise output levels, while the DAC used in this two step subranging like ADC only has a few coarse output levels. But these output levels must be reproducible at least as accurate as the overall ADC resolution.

Second, the dynamic requirements are different. The output of the internal DAC is used for comparison or subtraction at certain moments. Hence, the nonlinear transient behaviour like glitches is not important as long as the output levels are accurate at those moments. But the glitches will deteriorate the spurious free dynamic range of a stand alone DAC.

Third, the loading is different. A stand alone DAC may drive large capacitive (pF) and resistive loads so the buffer stage at the output is often required. The internal DACs are often followed by comparators or subtractors so the loading is usually capacitive and small. For a CMOS 65nm technology, the capacitive load is on the level of tens of fF.

Fourth, the measurement is different. For stand alone DACs, many samples are collected in one clock cycle and fourier-transformed to compute frequency related specifications (with the input codes of a sinusoidal signal). But for internal DACs, only the data at the sampling moment is taken and either discrete-time-fourier transformed to compute frequency related specifications or check if the samples are reproducible within the specified accuracy.

3.3 Speed-limiting factors

The internal DAC is designed for high speed ADC so it is important to understand the fundamental speed-limiting factors in internal DACs. The fore-going discussion only treats one limiting factor at a time. In a real internal DAC, these factors could act simultaneously, and thus result in compounding effects difficult for the analysis.

3.3.1 MOSFET channel set-up time

In order for NMOS/PMOS switches to function, the inversion layer at the MOSFET surface has to be formed first. The majority of carriers in the channel are pumped from the drain/source region. This channel set-up time defines the fastest speed that DAC can run without considering other delays. The velocity of electrons depends on many factors, such as, bulk doping, strength of electric field at the surface, and the quality of Si-SiO_2. But the electron saturation velocity offers the possibility of estimating the channel set-up time.

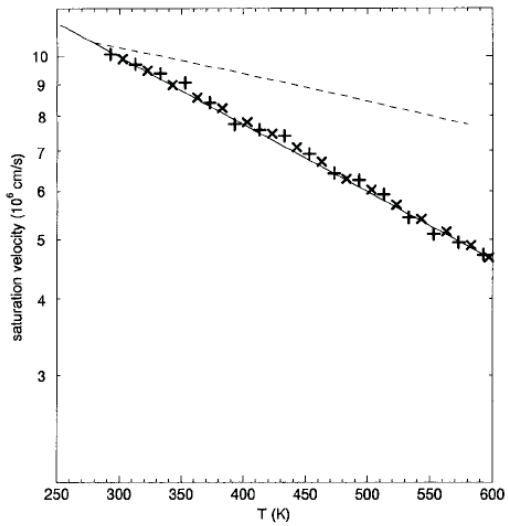


Fig. 3.3 Electron saturation velocity versus temperature [5]

The channel set-up time is estimated by the switch channel length divided by the electron saturation velocity.

$$T_{setup} = L / v_{sat} \qquad \text{(e3.1)}$$

Therefore, shorter the channel length is, faster the switches function. The following table is generated to demonstrate the shortest set-up time for different technologies. 8×10^6cm/s is used for the computation.

Table 3.1 Technology versus channel set-up time

Technology (μm)	Channel set-up time (ps)
0.12	1.5
0.09	1.125
0.065	0.8125

From the table above, the channel set-up time is on the order of pico seconds so this is not the dominant speed limiting factor.

3.3.2 RC settling time

There are many cases where the RC settling time becomes the speed limitation. The resistor could be the equivalent resistance from the resistor string or the switch on-resistance in capacitor DAC, and the capacitor could be the gate capacitance of subtractor or a capacitor component in the circuit. This settling behavior can be modeled as the step response of a resistor in series with a capacitor.

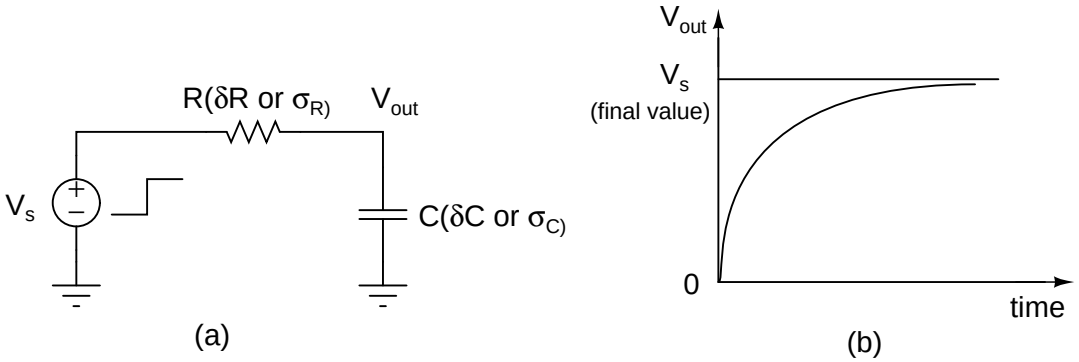


Fig. 3.4 (a) Low pass filter; (b) Its step response

The step response of the low pass filter is:

$$V_{out}(t) = V_s\left(1 - e^{-t/RC}\right) \qquad \text{(e3.2)}$$

where V_s is the final value. The exponential term, $V_s e^{-t/RC}$, is the dynamic settling error. This error approaches zero as the time passes by. The product of R and C is called time constant, often designated as τ. For the resolution of N bits, the error should be less than 1LSB.

$$e^{-t/RC} \le 2^{-N} \text{ or } t \ge N \cdot \ln 2 \cdot RC \qquad \text{(e3.3)}$$

Eqn. 3.3 specifies the minimum required settling time to reach N-bit accuracy for the nominal values of R and C.

The resistance and capacitance may not be the nominal value, R and C, due to the stochastic nature in the fabrication. Therefore, the settling time should be chosen to be long enough for the largest time constant to reach N-bit accuracy. For the random error in R and C, the required minimum settling time to reach N-bit accuracy is calculated from eqn. 3.4 [Appendix C-3].

$$e^{-\frac{t}{RC}} \cdot \left[1 + \frac{3t}{RC} \cdot \sqrt{\left(\frac{\sigma_R}{R}\right)^2 + \left(\frac{\sigma_C}{C}\right)^2}\right] < 2^{-N} \qquad \text{(e3.4)}$$

The equation above is derived for linear resistor and capacitor. The table below presents the minimum required time in terms of nominal time constant (τ = RC) to reach the N-bit resolution.

Table 3.2 The settling time versus resolution

Resolution (bits)	# of nominal τ (eqn. 3.3)	# of nominal τ due to random error (eqn. 3.4)
10	6.93	7.13
11	7.62	7.84
12	8.32	8.54
13	9.01	9.25
14	9.7	9.94

Note: Data are obtained with 1% error for $\frac{\sigma_R}{R}$ and 0.1% for $\frac{\sigma_C}{C}$.

As seen from the table, the required minimum number of time constant becomes larger with the presence of random and deterministic errors in R and C and increases with these errors.

3.4 Reproducibility-limiting factors

The internal DAC is also required to have high reproducibility so it is important to understand the common reproducibility-limiting factors in internal DACs. The foregoing discussion is the same as that for speed-limiting factors, treating one limiting factor at a time as if other factors did not exist. Equations derived are based on linear RC model.

3.4.1 Memory effect

Capacitors are the most common seen memory elements in internal DACs. The nodes in association with capacitors have different states for each output. There could be situation that the internal DAC switches from different codes to the same code. This means that the circuit converges to the same state with different initial conditions. If given enough settling time, the circuit will converge to the same state. But if not, the circuit states will be slightly dependent on the initial condition. The different initial conditions are determined by the code preceding the current settling. Fig. 3.5 illustrates the situation. The output settles to the final value, V_{out}, from two different initial conditions. One is from V_1 above V_{out} and the other from V_2 below V_{out}. ΔV represents the reproducible error at the sampling moment.

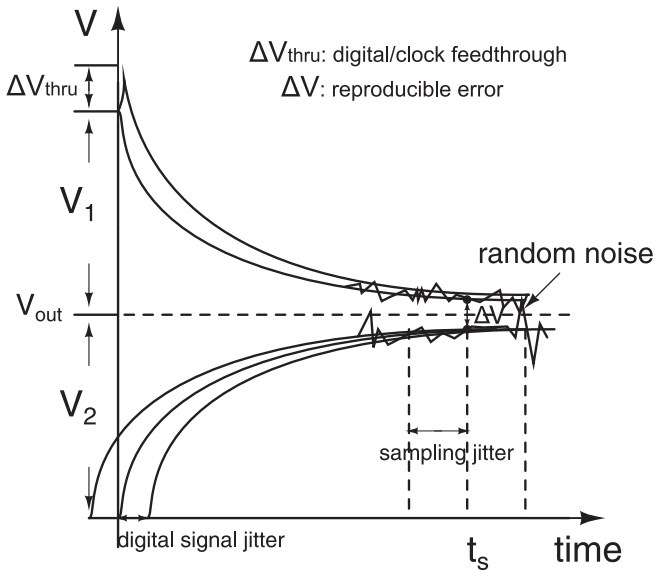


Fig. 3.5 The settling behaviour of DAC output

The reproducible error is defined as

$$\Delta V = (V_1 + V_2)e^{-\frac{t_s}{RC}} < V_{FS}e^{-\frac{t_s}{RC}} \qquad \text{(e3.5)}$$

where V_{FS}: the full scale DAC output, t_s: the sampling moment, RC: the equivalent time constant for the settling curves. The reproducible error varies with different V_1 and V_2. The largest reproducible error occurs when $V_1 + V_2 = V_{FS}$.

3.4.2 Signal jitter

The jitters influence the reproducibility by effectively decreasing or increasing the settling time. This is also pictured in Fig. 3.5. The concerned jitter in the project is the period jitter (a clock's output transition from its ideal position) because the DAC output needs to settle within the accuracy in the absolute time of 500ps. The figure shows two jitters: the digital signal jitter and the sampling clock jitter. The digital signal jitter is associated with the digital command to start the new settling in the internal DAC. The sampling jitter is associated with the sampling clock. It is seen from the figure that the largest reproducible error happens when the digital signals are delayed and the sampling moment is moved ahead.

$$\Delta V = (V_1 + V_2)e^{-\frac{t_s - 3\sqrt{2}\sigma_{clk}}{RC}} < V_{FS}e^{-\frac{t_s - 3\sqrt{2}\sigma_{clk}}{RC}} \qquad \text{(e3.6)}$$

where σ_{clk}: digital signal and sampling clock jitter, the rest notations have the same definition as in eqn. 3.5. The equation assumes that the digital signal jitter is equal to the sampling clock jitter, and they are uncorrelated. The minimum required settling time to reach N-bit resolution becomes

$$t_s > N \cdot \ln 2 \cdot RC + 3\sqrt{2}\sigma_{clk} \qquad \text{(e3.7)}$$

Another way to read this equation is that RC constant is determined by the jitter and accuracy requirements and the maximum allowed settling time in internal DACs (usually a clock cycle).

One clock source is shared among the functional blocks in ADC so the digital signal jitters and the sampling clock jitter are related to the absolute jitter at the clock transitions. The figure below shows the relationship between these jitters.

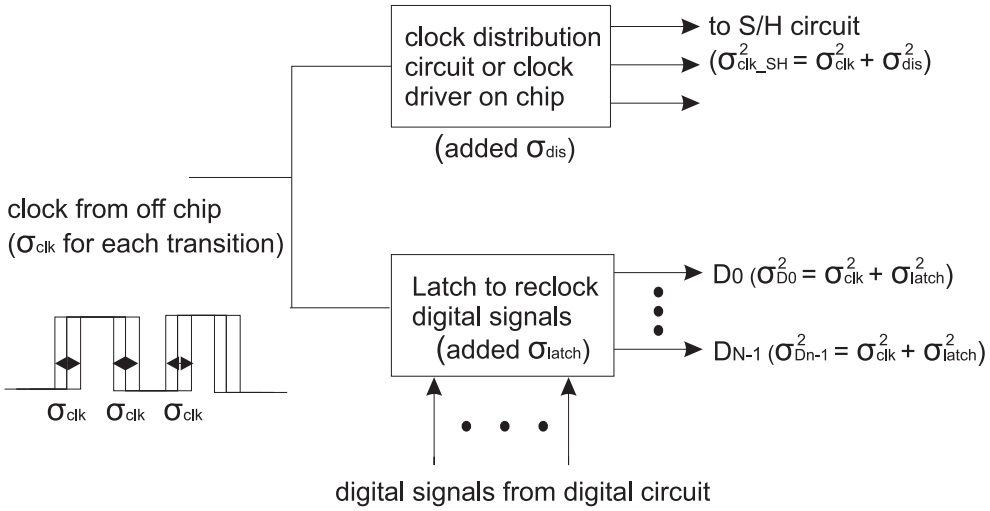


Fig. 3.6 Clocking in the system

Digital signals are reclocked by the latches and the sampling clock comes from the clock driver or redistribution network. There are added jitters, σ_{dis} and σ_{latch}, to the

clock jitter from off chip. In this chapter, it is assumed that $\sigma_{dis} = \sigma_{latch} = 0$, all the digital signal jitters are fully correlated, the digital signal jitters and the sampling clock jitter are fully uncorrelated and equal to σ_{clk} because DAC output updating and sampling take place at different transitions. The relationship between clock period jitter and σ_{clk} is

$$\sigma_{clk} = \frac{\sqrt{2}}{2}\sigma_{period} \tag{e3.8}$$

where σ_{period} is the period jitter for the clock from off chip.

3.4.3 Digital signal/clock feedthrough

The internal DACs are controlled by the digital signals or the clock. There are sharp transitions in these signals. During the transition moments, the peaks appear at the DAC output due to the parasitic coupling (refer to Fig. 2.3). This behaviour is equivalent to modifying the initial conditions at the beginning, settling from a larger initial value and the reproducible error increases by $\Delta V_{thru} e^{-\frac{t_s}{RC}}$. Accordingly, the minimum required number of time constant is increased.

$$e^{-\frac{t}{RC}} \cdot \left[1 + \frac{3t}{RC} \cdot \sqrt{\left(\frac{\sigma_R}{R}\right)^2 + \left(\frac{\sigma_C}{C}\right)^2}\right] < \frac{1}{1+x} \cdot 2^{-N} \tag{e3.9}$$

where x is the magnitude of digital feedthrough normalized to the full scale output range. The magnitude of feedthrough is reduced by choosing small switch size. The table below shows the minimum required time constant with the presence of both mismatch and digital feedthrough for 13-bit resolution.

Table 3.3 The settling time versus different magnitude of feedthrough

x (magnitude of digital feedthrough normalized to V_{FS})	# of nominal τ (eqn. 3.3)	# of nominal τ (eqn. 3.9)
25%	9.01	9.49
50%	9.01	9.67

Note: Data are obtained with 1% error for $\frac{\sigma_R}{R}$ and 0.1% for $\frac{\sigma_C}{C}$.

3.4.4 Switch control signals

Switches in the internal DACs are controlled by digital signals or clock signal. Switches usually go into saturation first and then fully turned on after control signals almost rise/fall to their final value. Due to charging/discharging switch gate capacitance, it is equivalent that the output settling is delayed by approximately rise/fall time of control signals.

3.4.5 Noise

Noise is stochastic in nature. Fig. 3.5 shows how it deteriorates the reproducibility. The root-mean-square (rms) noise on the sampling capacitor should be much less than 13-bit resolution. The noise requirement is dependent on other error sources.

$$\left(V_{FS} + \Delta V_{thru}\right) e^{-\frac{t_s - 3\sqrt{2}\sigma_{clk} - \alpha \cdot t_{r/f}}{RC}} + 6\sigma_{n\_out} < V_{1LSB@13-bit} \tag{e3.10}$$

where $t_{r/f}$: rise or fall time of a transition in switch control signals, α: the number of delay for the start of the settling normalized to rise or fall time, σ_{n_out}: the output noise at the sampling moment. The rest notations are the same as before. The equation above treats the different mechanisms separately without considering the interations between them.

3.5 Internal DAC design considerations

There are a few issues regarding the internal DAC design, such as whether a bootstrap circuit or the complementary switch should be used in the design, what type of load the internal DAC is generally driving, etc. Through defining design guidelines, they provide the same boundary conditions for easy comparisons between different topologies in terms of speed, reproducibility, and power consumption.

3.5.1 Switch considerations

MOSFETs are often used as switches in CMOS circuits. This is because MOSFETs operating in deep triode region exhibit no dc shift between the input and output voltages. There are three known topologies, simple NMOS or PMOS switch, the complementary switch (NMOS & PMOS pair) and NMOS/PMOS with the bootstrap circuit.

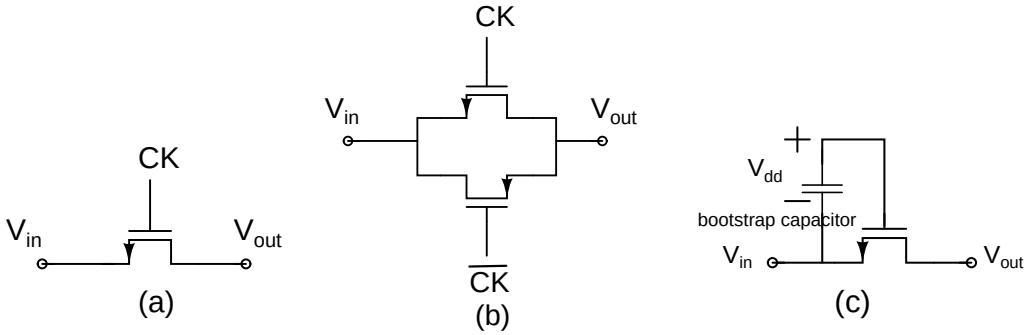


Fig. 3.7 (a) NMOS switch; (b) complementary switch; (c) boosted switch

These swich topologies are shown in Fig. 3.7. The switch is used in DACs to connect the physical quantity of different weights, voltage/charge/current, to the output.

The pros and cons of different switch topologies in CMOS circuits are summarized in the following table for later reference.

Table 3.4 Pros and cons of different switch topologies

Switch topology	Advantages	Disadvantages
Single transistor	Simplicity	Input-dependent on resistance; NMOS/PMOS acts as switch for certain input range.
Complementary switch (or transmission gate)	Relative low and input independent on resistance; medium complexity for switch driving	Acts as switch for certain input range; the on resistance is not small for technologies with low supply voltage.
Switch with bootstrap circuit	Input independent on resistance; very low on resistance even for low supply voltage	Bootstrap circuit required; reliability issue for the switch, large chip area due to the boostrap capacitor.

The switch with the bootstrap technique will not be used in the design because

- Each bootstrap circuit has about 3pF capacitance so the total capacitance grows exponentially with the DAC resolution, which means large chip area and power consumption at high speed.
- The maximally allowed settling time is half clock cycle as compared to the full clock cycle since the bootstrap circuit needs half clock cycle to charge the bootstrap capacitor to the supply.
- If boosted switch is used, digital control function needs to be incorporated in order to select the desired output level. The DAC output settling starts at least two times later than one rising/falling time. This is because the switch related to digital signal and the boosted switch are both needed to be turned on and it happens in serial time order. This even reduces the settling time far below half clock cycle.

3.5.2 Loading

The internal DAC provides output values to the subtractor in the two-step subranging like ADC. The differential pair is well known to implement the subtraction function. In other ADC topologies, internal DAC output values are fed to the comparators for comparison. It is reasonable to assume the gate capacitance of subtractor or comparator as the load to the internal DAC for a general discussion. Therefore, the internal DAC study in this chapter does not only serve the internal DAC design in the project, but also lends itself to the internal DAC design in other ADC topologies. The load capacitor is set to be 100fF for different designs since CMOS 65nm technology is used and the transistor gate capacitance is usually smaller than that.

3.5.3 Noise requirement

Generally, the internal DACs can be modeled as an active noise source, including thermal and 1/f noise components, connecting to a resistor and a capacitor in series. The active noise source depends on active circuit topology. The active noise source is uncorrelated to the resistor thermal noise.

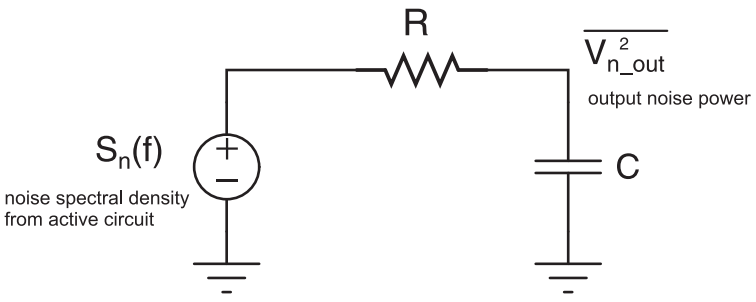


Fig. 3.8 Noise model for internal DAC

The rms total integrated output noise voltage across the load capacitor is

$$\sqrt{\overline{V_{n\_out}^2}} = \sqrt{\int_0^\infty \left(\underbrace{\frac{4kTR}{1+(2\pi f \cdot RC)^2}}_{\text{resistor thermal noise}} + \underbrace{\frac{S_n(f)}{1+(2\pi f \cdot RC)^2}}_{\text{active noise}} \right) df} = \sqrt{(1+\alpha)\frac{kT}{C}} \qquad \text{(e3.11)}$$

where kT: the thermal energy, R: the equivalent resistance, *C*: the load capacitance, α: factor due to the active noise source. Even though the DAC output noise with 100fF load capacitor could be larger than 13-bit resolution, the noise at the sampling moments can be suppressed to the desired level as long as the hold capacitance is sufficiently large.

3.5.4 Component matching consideration

As discussed in section 3.1, the component matching is not critical in the design so the discussion for different internal DAC designs will mainly focus on the effect of the dynamic error on the output reproducibility.

3.5.5 Simulation measurement

The measurement for the internal DAC in the project is whether the output levels are reproducible with the required accuracy. The testing input codes are randomly generated so that all the possible transitions are included for each level (Fig. 3.8 (a)) and the input codes should repeat at least two times in a row.

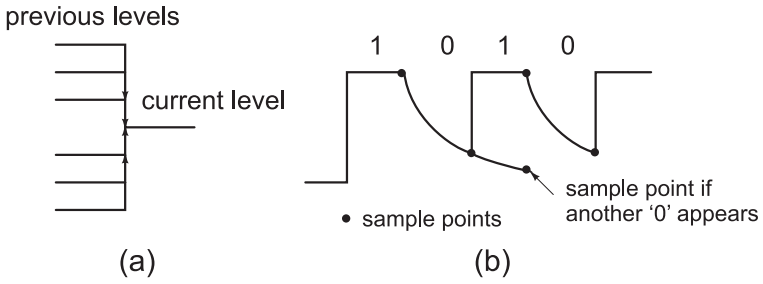


Fig. 3.9 Testing (a) transitions from different levels; (b) repetitive codes

Fig. 3.9 (b) depicts one situation: the reproducible error for '0's still satisfies the requirement without complete settling if the code repeats with '1010…' and the output reaches the equilibrium. Therefore, each input code should repeat at least two

times in a row to avoid the described situation. The output levels are collected at the sampling moments. All the collected data for each level should be centered around the average within the required accuracy. A sinusoidal input is not a good option because levels only 'see' fixed codes in front of them so the input with different frequencies should be used. The frequencies are chosen so that the testing requirements mentioned above are met. The proposed testing with random code generator running for hundreds of cycles requires less work and is simpler so it will be used in the study.

To estimate the allowed jitter in the system, the sampling moments are moved Δt ahead of the end of settling time until the output reproducibility is worse than 13-bit resolution. The obtained Δt is used in the jitter estimation. The allowed settling time is a clock cycle or half clock cycle dependent on the internal DAC implementation. The optimum sampling moment is in the middle of Δt as pictured below.

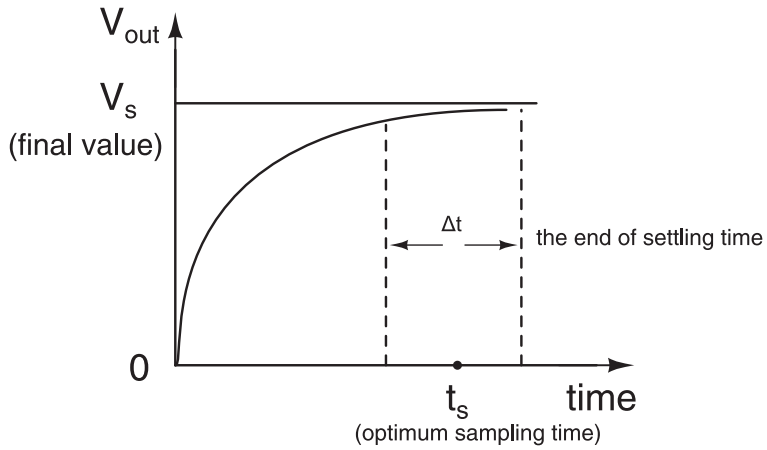


Fig. 3. 10 Jitter requirement estimation model

The jitter requirement is calculated by

$$\Delta t = 6\sqrt{2}\sigma_{clk} \qquad \text{(e3.12)}$$

In addition to the measurements above, the output noise at the sampling moments should be within the required accuracy. This can be checked by the transient noise simulation in cadence with large clock cycle. The output settles with sufficient time and only the noise is the cause of irreproducibility at the output. If the sampled outputs are reproducible with the required accuracy, then the noise meets the specification.

3.6 Design of different internal DACs

The internal DACs share the similar architectures as the stand alone DACs. This section studies the core part of different internal DACs to find out their speed, reproducibility limitations and power consumption. For each architecture, the discussion is divided into six subsections.

a. *General considerations* describe the operation mechanism and second order effects like digital feedthrough in different topologies.

b. *Component consideration* derives the design equation to meet the speed and reproducibility specifications.
c. *Switch consideration* discusses the switch choice and its effect on the DAC performance.
d. *Power consumption estimation* presents the static and dynamic power expressions for the internal DAC core.
e. *Reproducibility discussion* summarizes the possible causes to the reproducibility.
f. *Simulation data and explanations* presents the key component values in the design and the simulation data, mainly Δt, output noise, and power consumption. The explanations on the deviation from the ideal evaluation are also given.

The same design specifications are used for different internal DACs: 3-bit resolution and 13-bit reproducibility for single ended output excluding the noise at 2GS/s and 7.1ps allowed clock period jitter (σ_{clk} = 5ps, refer to eqn. 3.8) or $\Delta t \geq 42$ps. Performances are compared in aspects, such as, noise, power consumption, etc. Component mismatch is not included in the simulations. Higher resolution and the same reproducibility can be achieved by proper scaling. To isolate the effect from other circuits and focus on internal DAC study only, the following assumptions are made in the study.

- Supply is used as the reference voltage in all the designs. It has noise level much lower than 13-bit resolution and always provides sufficient current by connecting enough decoupling capacitor.
- One clock source with 50% duty cycle is shared in the system. The reclocking circuit and clock distribution network or driver do not add extra jitter, all the digital signal jitters are fully correlated, the digital signal jitters and the sampling clock jitter are fully uncorrelated and equal.
- To simplify capacitive dynamic power consumption derivation, the load capacitor or capacitor array in capacitor DAC is charged to an output level first and then discharged to ground in each clock cycle. Therefore, the estimation is for the largest dynamic power consumption.
- The load capacitor is set to be 100fF for different designs.
- Each digital signal from the reclocking circuitry can drive 20fF load capacitor with ~50ps rise and fall time (obtainable driving capability from digital circuit). This condition is assimilated by an ideal signal generator with 2ps rise/fall time followed by a resistor of 1.2kΩ.
- The minimum required number of time constant is calculated without the consideration of component spread in the general discussion. But it is taken into account in the final design for the project.

3.6.1 Resistor string DAC with walking one decoding

The resistor string with walking one decoding is given attention here because the time constants of resistor string DAC with different decoding (binary vs. walking one) are almost equal (refer to Fig. 2.5) at 3-bit resolution, the resistor string with walking one decoding has less switches and potentially reduces the power consumption due to less switch drivers.

a. General considerations

The operation of this DAC has been discussed in section 2.4.1. The output is equal to the voltage at the chosen resistor node after sufficient settling time.

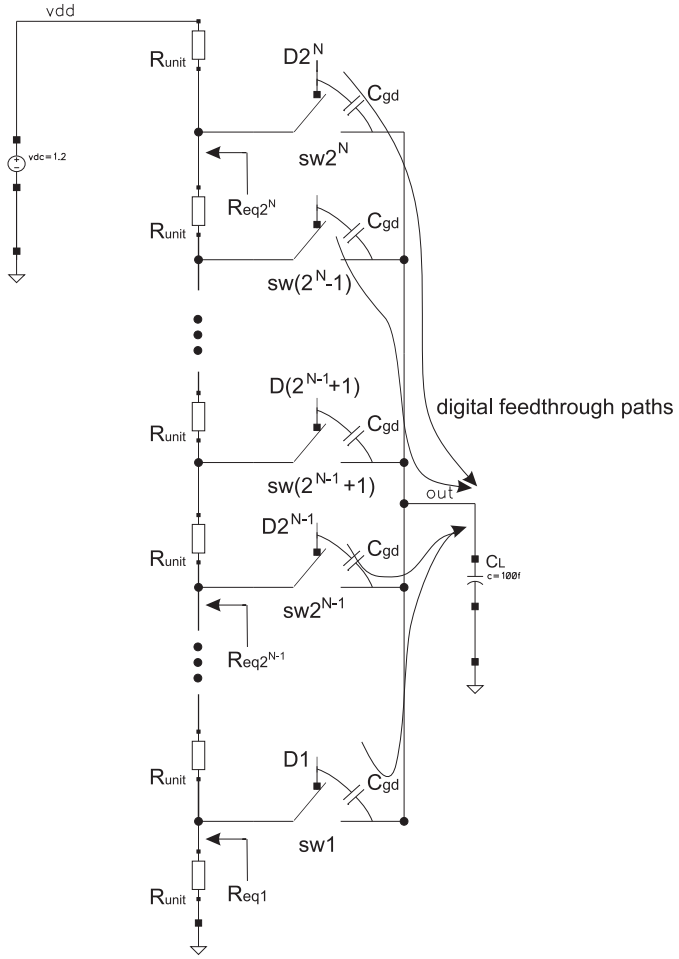


Fig. 3.11 Resistor string DAC with walking one decoding

N-bit DAC is composed of 2^N+1 unit resistors and each node in the resistor string connects to the output via a digitally controlled switch. The operation is similar to the sampling process. The sampled inputs are the different constant voltages defined by the resistor ladder. The maximally allowed settling time is one clock cycle. The settled output is

$$V_{out} = \sum_{n=1}^{2^N} D_n \cdot \frac{n}{2^N + 1} \cdot V_{DD} \qquad \text{(e3.13)}$$

There are digital feedthroughs to the output through the switch overlap capacitance. The largest feedthrough happens when the voltages in the middle tap are selected (refer to *switch consideration*).

$$\Delta V_{thru} = V_{DD} \cdot \frac{C_{gd}}{C_{L\_tot} - C_{gd}} \qquad \text{(e3.14)}$$

where C_{gd}: overlap capacitance of the switch that is turned on, C_{L_tot}: total output capacitance. The rms output noise is obviously equal to $\sqrt{KT / C_{L\_tot}}$.

b. Resistor consideration

Unit resistance is chosen so that different output levels can settle within the accuracy in a clock cycle. For each output level, the time constant could be different and equal to $(R_{eqn} + R_{swn}) \cdot C_{L\_tot}$, where R_{eqn}: the equivalent ladder output resistance for n^{th} level, R_{swn}: n^{th} switch on-resistance. R_{eqn} reaches maximum in the middle and R_{swn} is usually largest in the middle due to low overdrive voltage (refer to *switch consideration*). As a result, the DAC reproducibility can meet the specifications as long as the paths in the middle meet the following specifications.

$$t_{settling} > 6\sqrt{2}\sigma_{clk} + t_{r/f} + \beta \cdot (R_{eq\_\max} + R_{sw\_\max}) \cdot C_{L\_tot} \qquad \text{(e3.15)}$$

where $t_{settling}$: allowed settling time (500ps), σ_{clk}: clock transition jitter (5ps), $t_{r/f}$: rise/fall time of digital control signal dependent on the driving circuit, C_{L_tot}: total output capacitance, R_{sw_max}: maximum switch on-resistance, β: constant related to the accuracy, the magnitude of digital feedthrough and component mismatch.

$$R_{eq\_\max} = 2^{N-1} R_{unit} // (2^{N-1} + 1) R_{unit} = \frac{2^{N-1} \cdot (2^{N-1} + 1)}{2^N + 1} R_{unit} \qquad \text{(e3.16)}$$

β is ideally nine times the nominal time constant according to table 3.3. Ten can be chosen in the design to allow some room for the stochastic error and digital feedthrough (refer to section 3.4.3). One rise/fall time is used in the equation because the switch control signal needs to rise/fall (dependent on the type of switch, NMOS/PMOS) to one threshold voltage below/above the node voltage in order for switch to conduct assuming the worst initial output voltage.

c. Switch consideration

For the switches connecting to the node voltages close to the supply, PMOS is a good choice. Similarly, NMOS is a good choice for the switches connecting to the node voltages close to the ground. This can be seen from the overdrive voltage plot below.

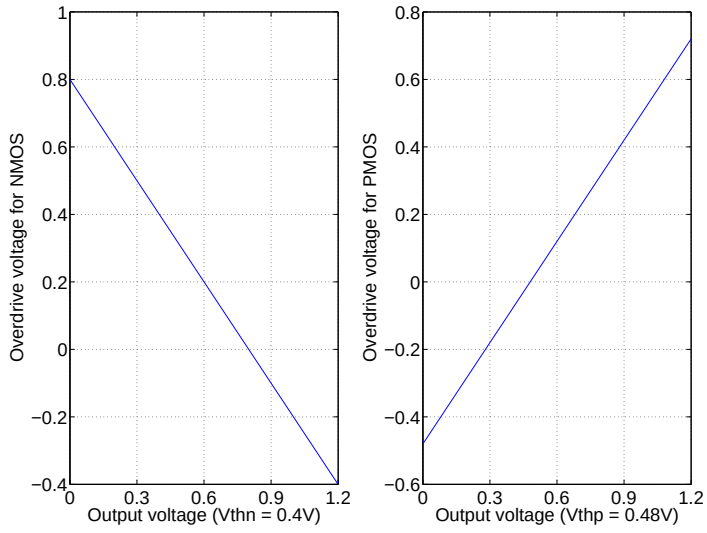


Fig. 3.12 Overdrive voltage for NMOS and PMOS

For the switches in the middle range, NMOS switch is chosen for the output voltage below half of the supply and PMOS chosen for the output voltage above half of the supply. Complementary switch is not chosen due to more driving circuits and possible higher power consumption. It is expected that the switch size is largest in the middle to achieve the same on-resistance because of low overdrive voltage. Therefore, the

switch gate-drain/source overlap capacitance reaches maximum in the middle and the consequence is that the maximum digital feedthrough is observed when the switches in the middle are turned on.

d. Power consumption estimation

The power consumption consists of DC and dynamic portions. DC dissipation is mainly due to the static current flowing through the resistor ladder. Large unit resistance is preferred to reduce DC dissipation. Dynamic power consumption is composed of that from charging/discharging the load capacitor and estimated by assuming that each code appears equally in terms of probability, $\frac{1}{2^N}$.

$$P_{total} = P_{DC} + P_{load} + P_{sw\_driver} \quad \text{(e3.17)}$$

where P_{sw_driver}: power consumption for switch drivers (estimated in the simulation),

$$P_{DC} = \frac{V_{DD}^2}{(2^N + 1)R_{unit}}$$

$$P_{load} = \sum_{n=0}^{2^N-1} C_{L\_tot} V_n^2 \cdot f_{clk} \cdot \frac{1}{2^N}$$

where N: the number of bits, R_{unit}: unit resistance, V_{DD}: the supply voltage, C_{L_tot}: total load capacitance, f_{clk}: clock frequency.

e. Reproducibility discussion

The reproducibility limiting factors are memory effect, digital signal jitter, finite transition time of digital signals and digital feedthrough to the output. All the factors influence the reproducibility as the way explained in section 3.4.

f. Simulation data and explanations

A three-bit resistor string DAC is designed and simulated in cadence to meet the design specifications. The unit resistance and switch sizes are chosen so that eqn. 3.15 is satisfied. The switch drivers, basically inverters, are designed to achieve less than 40ps rise/fall time ($t_{r/f}$) in digital signals. The random walking one code is generated by a verilog-A script [Appendix C-4]. With the cadence script [Appendix C-5], the simulation data are first collected at the end of clock cycle and computed for the largest difference as the reproducible error using Matlab code [Appendix C-6]. Then the multiple simulations are redone at different Δt ahead of t_s until the reproducible error reaches the requirement in the project. The jitter requirement is calculated from Δt according to eqn. 3.12. The silicided N+ poly resistor is chosen for the resistor ladder because of low resistance per square and small area to achieve low resistance.

Below are the design values for unit resistor and switches.

R_{unit} (Ω) and size (μm)	10.34; 0.24/0.16
sw1, R_{sw1} (Ω) and size W/L (μm)	36; 20/0.06 (NMOS)
sw2, R_{sw2} (Ω) and size W/L (μm)	56; 20/0.06 (NMOS)
sw3, R_{sw3} (Ω) and size W/L (μm)	42; 55/0.06 (NMOS)
sw4, R_{sw4} (Ω) and size W/L (μm)	36; 150/0.06 (NMOS)
sw5, R_{sw5} (Ω) and size W/L (μm)	73; 126/0.06 (PMOS)
sw6, R_{sw6} (Ω) and size W/L (μm)	54; 75/0.06 (PMOS)
sw7, R_{sw7} (Ω) and size W/L (μm)	67; 35/0.06 (PMOS)
sw8, R_{sw8} (Ω) and size W/L (μm)	50; 35/0.06 (PMOS)

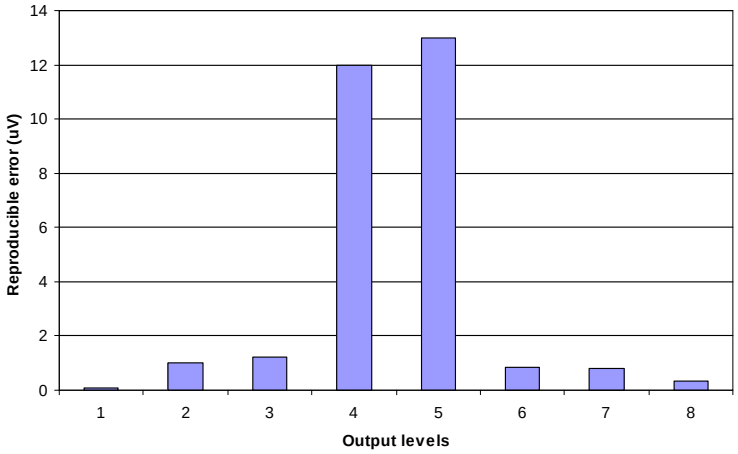


Fig. 3.13 Reproducible errors @ end of clock cycle versus different output levels

Below show the simulated and calculated performances.

	Δt /jitter requirement (ps)	rms noise (μV)	Total power consumption (mW)
Simulation	45/5.3	195	15.64
Calculation	281/33	143	15.49

Note: calculated total power does not include the power from switch drivers.

Below is the decomposition of total power consumption.

	Total power consumption (mW)	DC dissipation (mW)	Dynamic power from charging/discharging $C_{L\ tot}$ (mW)	Dynamic power from switch drivers (mW)
simulation	15.64	15.4 (98.5%)	0.068 (0.4%)	0.17 (1.1%)
calculation	15.49	15.4	0.091	x

It is seen that the switch sizes become larger in the middle due to low overdrive voltage and reproducible errors are also larger in the middle, as expected. The calculated output noise is a little lower than the simulated output noise because the noise model does not take into account 1/f noise. The estimated DAC DC power and dynamic power related to C_{L_tot} are close to the simulation data. DC power takes most power consumption.

There is a big difference between simulated Δt (45ps) and estimated Δt (281ps). The most critical path is via switch #5 and its time constant is equal to 19ps (95Ω·200fF) so the expected Δt is

$$\Delta t = 500ps - 40ps - 9.4 \times 19ps = 281ps$$

according to eqn. 3.15. β is 9.4 because there is about 20% voltage feedthrough in the design. The difference results from two extra sources: There is extra digital signal delay from the ideal generator to the switch#5 gate capacitor (period 1 in Fig. 3.14); the switch on-resistance is nonlinear during the settling process (period 3 in Fig. 3.14). These can be observed in the following graph.

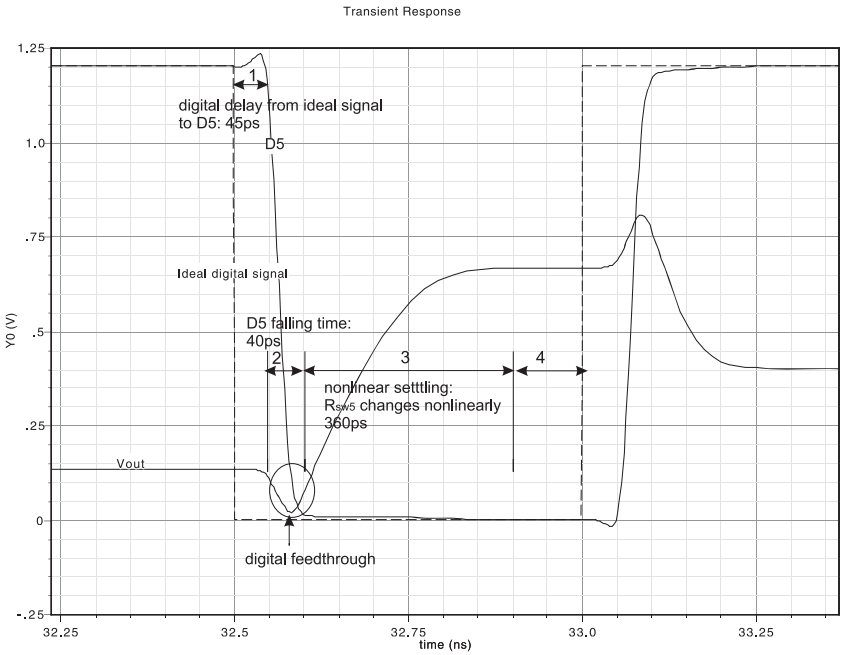


Fig. 3.14 Output settling via sw#5

When the ideal digital signal becomes low, the output is disturbed and loses its accuracy due to digital feedthrough so the digital delay from ideal signal source to the switch gate should be counted in the calculation. During period 3, sw#5 goes from saturation region to linear region. Its output resistance varies with V_{ds}. Below is the output resistance plot from Promost.

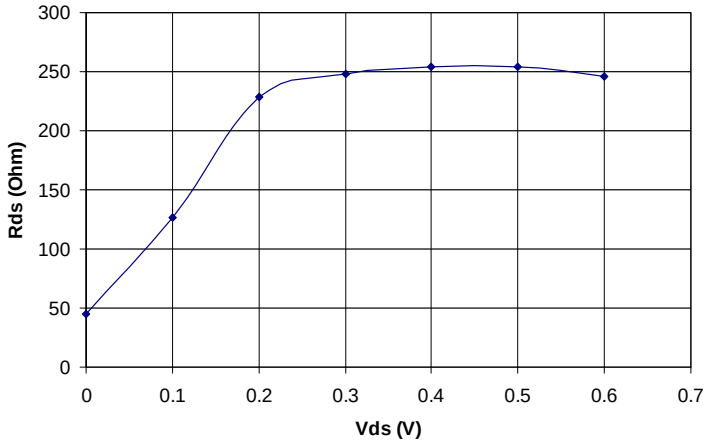


Fig. 3.15 output resistance versus V_{ds}

Therefore, it is expected that the settling takes longer time. If two thirds of highest value is used in the calculation, new Δt (62ps) becomes quite close to the simulation data. To ensure that the design can meet the specifications, the maximum on-resistance is suggested to be used in the equation.

3.6.2 R-2R ladder DAC

This section focuses on the voltage mode R-2R DAC because the current mode R-2R DAC needs OPAMP to provide virtual ground and transform the current to voltage at the output and thus reduce the maximum achievable speed.

a. General considerations

This type of internal DAC uses the inverse binary coding. The output for a N-bit DAC is described by

$$V_{out} = \overline{D}_N \cdot V_{DD}/2 + \overline{D}_{N-1} \cdot V_{DD}/4 + ... + \overline{D}_1 \cdot V_{DD}/2^N \qquad \text{(e3.18)}$$

where N is the number of bits, D_1 to D_N are the digital control bits.

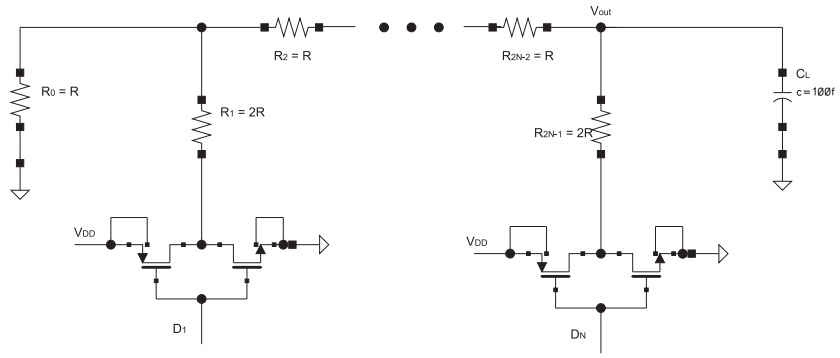


Fig. 3.16 R-2R ladder DAC with binary decoding

Fig. 3.16 shows N-bit R-2R ladder DAC with inverter switches. The 2R branches connect either to the supply or to the ground for each digital input code. Each bit generates a voltage weight at the output by resistor divider network and the output is the summation of these digitally controlled voltage weights. The output settles to the final value from the previous output level in one clock cycle. Digital feedthroughs in this topology are small because there are no direct capacitive paths from digital signals to the output and feedthroughs are attenuated by the resistive network. The rms output noise is approximately $\sqrt{KT/C_L}$.

b. Resistor considerations

There are only R and 2R resistors and the output resistance is equal to R. Their values are determined by the speed and the accuracy requirements. Below is the design equation to satisfy both the speed and accuracy requirements.

$$t_{settling} > 6\sqrt{2}\sigma_{clk} + \beta \cdot R \cdot C_L \qquad \text{(e3.19)}$$

where $t_{settling}$: the maximum allowed settling time (500ps), σ_{clk}: clock transition jitter (5ps), C_L: load capacitance, R: the equivalent output resistance, β: constant related to the accuracy and component mismatch. β is ideally nine times the nominal time constant according to table 3.3. Ten can be chosen in the design to allow some room for the stochastic error and digital feedthrough (refer to section 3.4.3). The rise/fall time is not included in the equation if the inverter output resistance keeps approximately constant during digital signal transitions and R_{on} is small as compared to R_N.

c. Switch consideration

The complementary switch is not used because it is similar to the inverter switch.

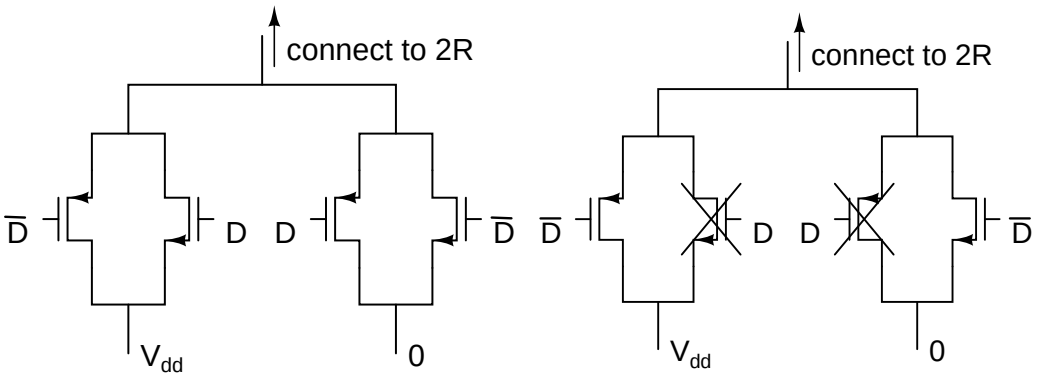


Fig. 3.19 The complementary switch

When D is high, the node on the top connects to the supply. NMOS on the left branch is not an effective switch because V_{gs} = 0. When D is low, the node on the top connects to the ground. PMOS on the right branch is not an effective switch. So the switch is similar to the inverter switch.

d. Power consumption estimation

The power consumption consists of DC and dynamic portions. For each code, different branches are connected to the supply voltage and the load capacitor is charged to the chosen value. Obviously, the dynamic power is code dependent. DC power is code dependent too since the equivalent resistance looking from the supply to the R-2R network is unequal for each code. This is explained by the following graph.

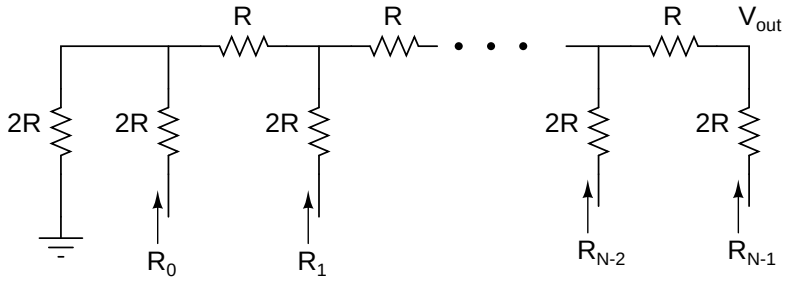


Fig. 3.20 The DC power estimation model

The resistance looking into each branch is designated as R_n with n from 0 to N–1. It is expressed by [Appendix C-7]:

$$R_n = 2R + \frac{4^{N-n}+2}{4^{N-n}-1}R, \quad n = 0, 1, 2, \ldots, N-1 \qquad \text{(e3.20)}$$

The equivalent resistance to the supply is all the resistances in parallel that are the branches with digital '1's. The total power consumption can be estimated by assuming that each code appears equally in terms of probability, $1/2^N$, and the load capacitor is charged to the final value from zero in each clock cycle.

$$\begin{aligned} P_{total} &= P_{DC} + P_{load} + P_{sw\_driver} \\ &= \alpha \cdot V_{DD}^2 / R + \beta \cdot C_L V_{DD}^2 \cdot f_{clk} + P_{sw\_driver} \end{aligned} \qquad \text{(e3.21)}$$

where

$$\alpha = \sum_{n=0}^{2^N-1} \frac{1}{R_{eq,n}} \cdot \frac{1}{2^N} \text{ and } \beta = \sum_{n=0}^{2^N-1} \left(\frac{n}{2^N} \right)^2 \frac{1}{2^N}$$

where N: the number of bits, $R_{eq,n}$: the equivalent resistance to the supply for different codes, V_n: the DAC output levels from 0 to $V_{DD} - V_{1LSB}$, f_{clk}: clock frequency, P_{sw_driver}: power consumption for switch drivers (estimated in the simulation). α and β are power coefficients related to DC and dynamic power consumption and independent of the supply voltage and the resistance. The Matlab program [Appendix C-8] is used to generate the coefficients for different resolutions, and they are summarized in the table below for later reference.

Table 3.6 DC and dynamic power coefficients

Resolution (bits)	α	β
3	0.4453	0.2734
4	0.6113	0.3028
5	0.7778	0.3178
6	0.9445	0.3256
7	1.1111	0.3294
8	1.2778	0.3314

e. Reproducibility discussion

The reproducibility limiting factors are mainly memory effect and digital signal jitter. These factors influence the reproducibility as the way explained in section 3.4.

f. Simulation data

A three-bit R-2R ladder DAC is designed and simulated in cadence to meet the design specifications. The resistance values are chosen so that eqn. 3.19 is satisfied. The switch drivers, basically inverters, are designed to achieve less than 40ps rise/fall time ($t_{r/f}$) in digital signals. The random binary code is generated by a verilog-A script similar to that in Appendix C-4. With the cadence script [Appendix C-5], the simulation data are first collected at the end of clock cycle and computed for the largest difference as the reproducible error using Matlab code [Appendix C-6]. Then the multiple simulations are redone at different Δt ahead of t_s until the reproducible error reaches the requirement in the project. The jitter requirement is calculated from Δt according to eqn. 3.12. The unsilicided resistor is chosen in the design because of large resistance per square.

Below are the design values for inverter sizes, R and 2R resistors. All the inverter switches have the same dimensions.

Inverter NMOS, R_{sw_nmos} (Ω) and W/L (μm)	19.3; 28/0.06
Inverter PMOS, R_{sw_pmos} (Ω) and W/L (μm)	19; 76/0.06 (NMOS)
R (Ω) and size W/L (μm)	456; 0.25/0.49
2R (Ω) and size W/L (μm)	903; 0.24/0.99

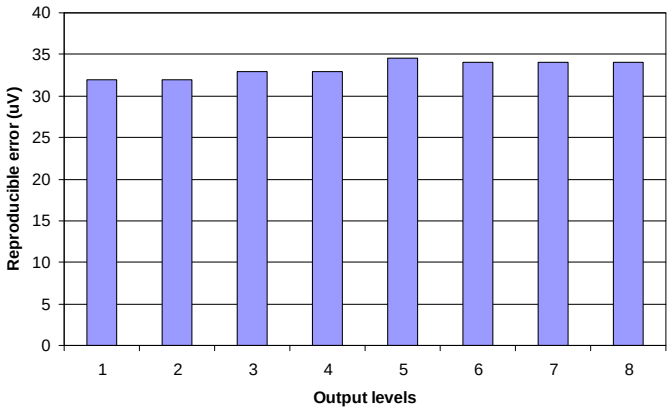


Fig. 3.21 Reproducible errors @ end of clock cycle versus different output levels

Below show the simulated and calculated performances.

	Δt /jitter requirement (ps)	rms noise (μV)	Total power consumption (mW)
Simulation	45/5.3	225	1.46
Calculation	54/6.4	202	1.68

Note: calculated total power does not include the power from switch drivers. Digital signal delay of 54ps is included in Δt estimation and β is 9.01 because of low digital feedthrough in this topology and no consideration of component spread.

Below is the decomposition of total power consumption.

	Total power consumption (mW)	DC dissipation (mW)	Dynamic power from charging/discharging C_L (mW)	Dynamic power from switch drivers (mW)
simulation	1.46	1.2(82%)	0.02(1.4%)	0.24 (16.6%)
calculation	1.68	1.4	0.08	x

The reproducible errors at different output levels are almost equal since the output resistances are almost constant at different levels. The calculated output noise is a little lower than the simulated output noise because the noise model does not take into account 1/f noise. The estimated DAC DC power and dynamic power related to C_L are close to the simulation data. DC power takes most power consumption.

3.6.3 Capacitor DAC

Capacitor DAC output can be defined in two ways. One is by charge sharing as shown in Fig. 2.7. The selected capacitors are first charged to the supply and then they share the charge with the capacitor array. It is required that the reset switch turns off before the charge sharing action. The other is by capacitor divider network as discussed in this section. Capacitor DAC with charge sharing mechanism is not discussed because effective settling time (output voltage defining time) is less than half clock cycle, the total reproducible error is the sum of errors in reset phase and charge sharing phase, and careful timing for reset clock is required, which effectively reduces the settling time. Capacitor DAC is widely used in ADC systems due to low power consumption and its capability of the integration with the subtraction and hold functions.

a. General considerations

The inverse binary coding is used since glitches are allowed in the internal DAC design; the control word from digital circuit is in binary form and thus no decoding circuit is required between digital circuit and the internal DAC; fewer switches are required as compared to using thermometer code and thus save chip area. The output for a N-bit DAC is described by

$$V_{out} = \sum_{n=1}^{N} \overline{D_n} \cdot V_{DD} \cdot \frac{C_n}{C_t + C_L} \qquad \text{(e3.22)}$$

where D_n: digital control bit, V_{DD}: supply voltage, C_L: load capacitance, C_t: $\sum_{n=1}^{N} C_n$, the total capacitances of different weights. Due to the load capacitor, each DAC output level deviates from its ideal level. The offset reaches maximum at the largest output level.

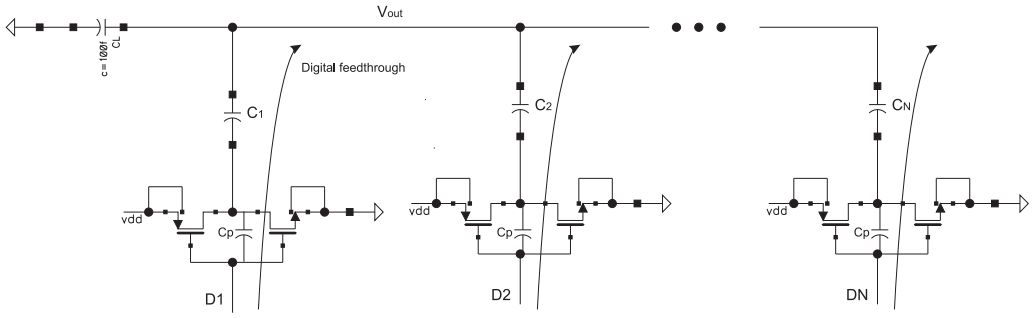


Fig. 3.22 Capacitor DAC with binary decoding

Fig. 3.22 shows an N-bit capacitor DAC with inverter switches. C_n connects either to the supply voltage or to the ground for each digital input code. Each bit generates a voltage weight at the output through capacitor divider network and the output is the summation of these digitally controlled voltage weights. The output settles to the final value from the previous output level in one clock cycle. There are digital feedthroughs from the digital bits to the output via the inverter switch gate capacitors. The magnitude of feedthrough is proportional to the number of bit changes and the switch size. The rms output noise is approximately $\sqrt{\frac{KT}{C_L + C_t} \cdot \sum_{n=1}^{N} \frac{C_n}{C_L + C_t - C_n}}$ [Appendix C-11].

b. Capacitor considerations

The total capacitance is chosen so that KT/C noise is below 13 bit level.

c. Switch considerations

The complementary switch is not used because it is similar to the inverter switch (refer to section 3.6.2 c). The switch on-resistance plays an important role in achieving the reproducible output levels. To find out the critical path in the design, equal size of inverter switches is used for different bits. This also avoids redesigning inverter switches at different bits as long as the critical path can settle within the accuracy. The maximum settling time of the DAC is usually determined by the MSB settling.

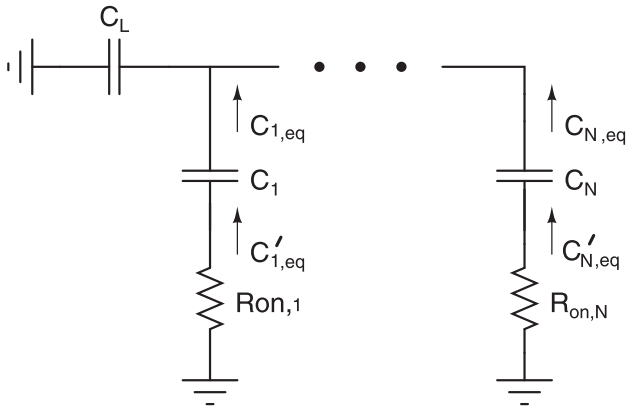


Fig. 3.23 RC settling estimation model

$R_{on,n}$ in the figure designates PMOS or NMOS switch on-resistance at n^{th} bit and they are equal (R_{on}), $C_{n,eq}$ is the equivalent series capacitance with C_n, with n from 1 to N.

$$C_{n,eq} = C_L + \sum_{i=1, i\neq n}^{N} C_i \tag{e3.23}$$

$$C'_{n,eq} = \frac{C_{n,eq} \cdot C_n}{C_{n,eq} + C_n} = \frac{(C_L + C_t - C_n) \cdot C_n}{C_t + C_L} \tag{e3.24}$$

where C_L: load capacitance, N: the number of bits, C_n: the capacitor at n^{th} bit, C_t: the total capacitance, $\sum_{n=1}^{N} C_n$. $C_{n,eq}$ is obtained by assuming zero switch on-resistance. $C'_{n,eq}$ increases with n and reaches maximum at N [Appendix C-9]. The switch sizes can be determined by [Appendix C-9]

$$t_{settling} > 6\sqrt{2}\sigma_{clk} + \beta \cdot \alpha \cdot R_{on} \cdot C'_{N,eq} \tag{e3.25}$$

where $t_{settling}$: the allowed settling time (500ps), σ_{clk}: clock transition jitter (5ps), R_{on}: the switch on-resistance, β: constant related to the accuracy and component mismatch, α: scaling factor related to R_{on}. α is between $1 + \frac{1}{2^{N-1} - 1}$ and 2 [Appendix C-9]. β is ideally nine times the nominal time constant according to table 3.3. Ten can be chosen in the design to allow some room for the stochastic error and digital feedthrough (refer to section 3.4.3). The rise/fall time is not included in the equation because the inverter output resistance keeps approximately constant during digital signal transitions and settling is not delayed.

d. Power consumption estimation

The power consumption only consists of dynamic dissipation.

$$P_{total} = P_{C\_array} + P_{sw\_driver} \tag{e3.26}$$

where P_{C_array}: the power dissipation related to charging/discharging the capacitor array C_n and C_L, P_{sw_driver}: power consumption for switch drivers (estimated in the simulation).

P_{C_array} is code dependent and can be estimated by assuming that each code appears equally in terms of probability and different capacitors are charged to the final value from zero in each period. For an input code, the energy stored in the capacitor array is

$$E_n = \sum_{i=0}^{N-1}\left(D_i \cdot \frac{1}{2} \cdot C_i \cdot V_n^2 + \overline{D_i} \cdot \frac{1}{2} \cdot C_i \cdot (V_{DD} - V_n)^2 \right) + \frac{1}{2} \cdot C_L \cdot V_n^2 \qquad \text{(e3.27)}$$

And each code has the probability of $\frac{1}{2^N}$, thus

$$P_{C\_array} = \sum_{n=0}^{2^N-1} 2 \cdot E_n \cdot \frac{1}{2^N} \cdot f_{clk} \qquad \text{(e3.28)}$$

where N: the number of bits; C_i: $\sum_{n=0}^{N-1} C_n$; V_n: the DAC output levels; f_{clk}: clock frequency. Factor of 2 implies that both charging and discharging capacitor array consume equal amount of energy.

e. Reproducibility discussion

The reproducibility limiting factors are memory effect, digital signal jitter and digital feedthrough to the output. All the factors influence the reproducibility as the way explained in section 3.4.

f. Simulation data

A three-bit capacitive DAC is designed and simulated in cadence to meet the design specifications. The switch drivers, basically inverters, are designed to achieve less than 40ps rise/fall time ($t_{r/f}$) in digital signals. The random binary code is generated by a verilog-A script [Appendix C-4]. With the cadence script [Appendix C-5], the simulation data are first collected at the end of clock cycle and computed for the largest difference as the reproducible error using Matlab code [Appendix C-6]. Then the multiple simulations are redone at different Δt ahead of t_s until the reproducible error reaches the requirement in the project. The jitter requirement is calculated from Δt according to eqn. 3.12.

Below are the design values for inverter sizes and total array capacitance. All the inverter switches have the same dimensions.

Inverter NMOS, R_{sw_nmos} (Ω) and W/L (μm)	102; 5.3/0.06
Inverter PMOS, R_{sw_pmos} (Ω) and W/L (μm)	102; 14.2/0.06 (NMOS)
$\sum_{n=0}^{N-1} C_n$, (pF)	1

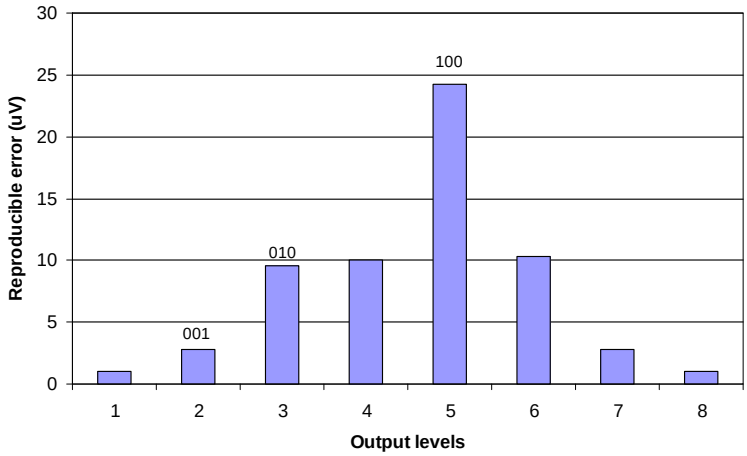


Fig. 3.24 Reproducible errors @ end of clock cycle versus different output levels

Below show the simulated and calculated performances.

	Δt /jitter requirement (ps)	rms noise (μV)	Total power consumption (mW)
Simulation	45/5.3	130	0.28
Calculation	48/5.7	81	1.2

Note: calculated total power does not include the power from switch drivers. Δt is calculated with α = 1.54 (between 1.33 and 2.0, as expected). Digital signal delay of 50ps is included in Δt estimation and β is 9.3 because there is about 10% digital feedthrough in this design and no consideration of component spread.

Below is the decomposition of total power consumption.

	Total power consumption (mW)	Dynamic power from charging/discharging C_L and capacitor array (mW)	Dynamic power from switch drivers (mW)
simulation	0.28	0.2 (71%)	0.08 (29%)
calculation	1.2	1.2	x

The reproducible error increases with the significance of bit and reaches maximum at the MSB, as expected. The calculated output noise is a little lower than the simulated output noise because the noise model does not take into account 1/f noise. The estimated DAC dynamic power related to capacitor array and C_L is much larger than the simulation data because the estimation is for the worse case.

3.6.4 Current steering DAC

The output of current steering DACs is in the form of current. The circuit is required to convert the output current to the voltage for the later signal processing. One solution is the transimpedance amplifier utilizing an OPAMP. This solution is not appropriate due to the speed limitation of the OPAMP. The other is the resistive load. The speed in this case is limited by the output pole. The second solution provides possible high speed design so the section will study the current steering DAC with the resistive load. NMOS transistors are chosen in the design because of faster electron mobility and less parasitics, which are desired charactistics in high speed applications.

a. General considerations

The binary coding is used because of the same reasons as those for capacitor DAC design: glitches are allowed and no decoding circuit is required. The output for a N-bit DAC is described by

$$V_{out} = \sum_{n=1}^{N} D_n \cdot I_n \cdot R_L \qquad \text{(e3.29)}$$

where D_n: the digital control bit, I_n: the current of different weights, R_L: the load resistance.

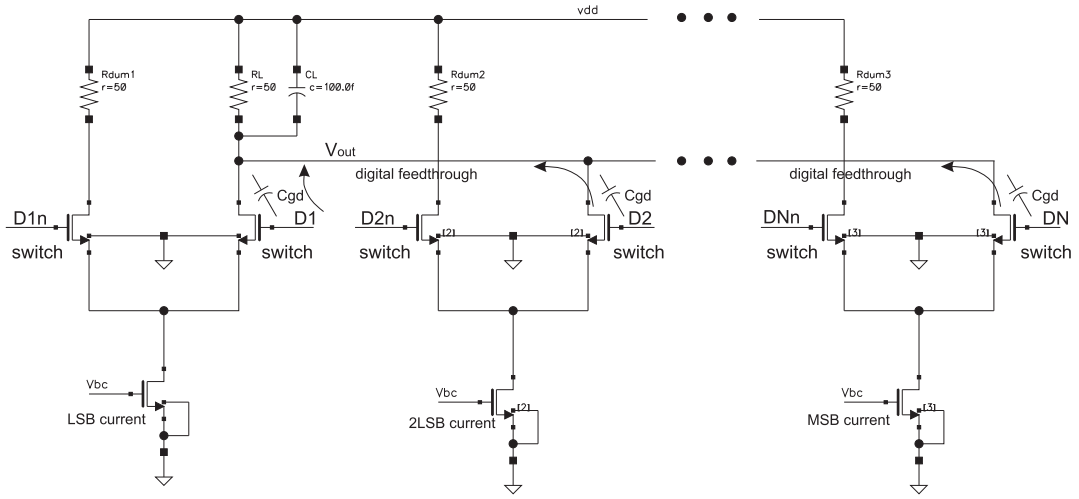


Fig. 3.25 3-bit current steering DAC and its output

The figure above shows an N-bit current steering DAC. The switches steer the currents of different weights to either the load or the dummy resistor. Each bit generates a voltage weight at the output by the load resistor, and the output is the summation of these digitally controlled voltage weights. The output settles to the final value from the previous code level in one clock cycle.

There are digital feedthroughs from the digital bits to the output. The schematic in Fig. 3.25 shows feedthrough paths to the output via the switch gate-drain overlap capacitance. The magnitude of feedthrough depends on the number of switchings so it is code dependent. Its effect is the modification of the initial conditions at the beginning of each settling action. The output noise is $\sqrt{(1+\alpha)\frac{kT}{C_L}}$, where α is the factor related to thermal and 1/f noise in the active circuit. α is code dependent and reaches maximum when all the currents are steered into the load. The output range of the topology is $[V_{DD} - I_{tot}R_L, V_{DD}]$ where I_{tot} is the total output current. The lower bound is set by the minimum required voltage to keep current source and the switch in saturation. The disadvantage for NMOS implementation is that the output can not be close to the ground level so PMOS implementation can be used instead.

b. Current cell considerations

The cascode configuration helps reduce the maximum offset in DAC output levels or INL [4], but it is not used in the design due to the limited voltage headroom in

standard CMOS 65nm process and the available calibration technique in ADC system to compensate the DAC output offset.

c. Switch considerations

There is a benefit of operating the switch in the saturation region. It is the reduced parasitic capacitance contribution to the output node, which is good to high speed applications. If the switch operates in the linear region, half the gate capacitance would add to the output as compared to much smaller gate-drain overlap capacitance.

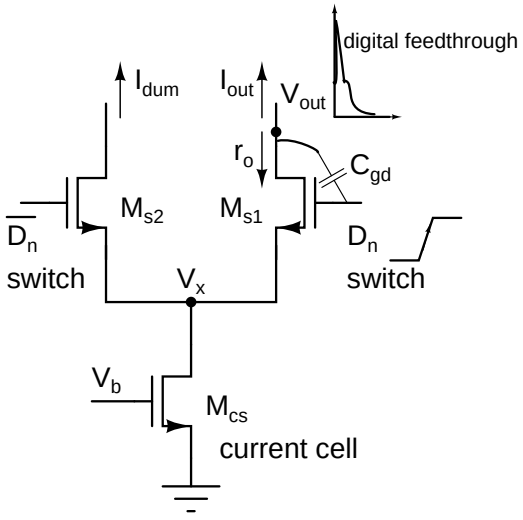


Fig. 3.26 A current cell with the switches

The digital feedthrough, coupling to the output via the switch gate-drain overlap capacitance, is a concern that affects the performance. The digital feedthrough results in an initial voltage step at the switching moment, as shown in Fig. 3.26. Its magnitude depends on the number of switching cells. The voltage variation at the output is approximately given by [6]

$$\Delta V \approx \frac{nC_{gd2}}{C_L + C_{dtot2}} \Delta V_{G2} \qquad \text{(e3.30)}$$

where ΔV_{G2}: the digital signal swing, nC_{gd2}: the total gate-drain capacitance of n switching transistors that are commuted simultaneously, V_{dtot2}: the total parasitic drain capacitance of the switching transistors, C_L: the load capacitance. The effect is reduced by the following measures. A transistor can be placed between the switch and the output isolating the direct feedthrough path. This is not appropriate due to the limited headroom. Another is to reduce the digital signal swing. i.e. 1V supply utilized for digital signals. The last is to use small width for the switch to lower the gate-drain overlap capacitance. The minimum length is chosen to keep the same W/L ratio. For the given DAC output swing, allowing large headroom for switch helps reducing the required width as well. Digital feedthrough is most pronounced when codes change from 00…0 to 11…1.

In the high speed operation, the switch pair should not be off at the same time during the digital transitions since charging/discharging the parasitic capacitance of the current source increases the settling time of the DAC. A driver circuit in [6] is used in the design to shift the crossing point of the switch transistor's differential control signals, in such a way that these transistors are never simultaneously in the off state.

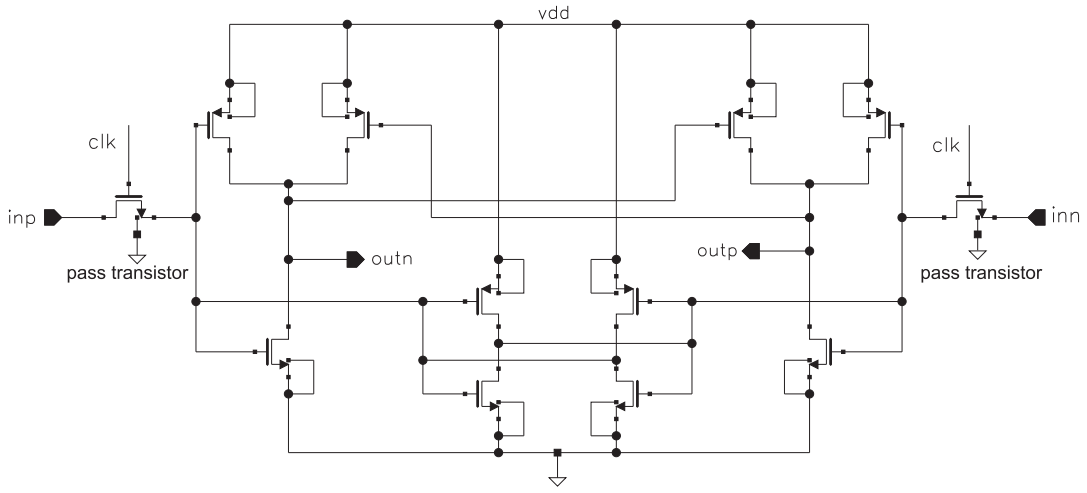


Fig. 3.27 The latch driver for the switches [6]

It is rise/fall-time based circuit. The crossing point is tuned by scaling the gate width of PMOS & NMOS transistors. The lower crossing point is achieved by decreasing NMOS gate width and increasing PMOS gate width. The higher cross point is achieved for the opposite. The positive feedback by the small inverters in the middle suppresses the clock feedthrough by the pass transistors and stabilizes the synchronized inputs. The latches are scaled up to drive the switches at the different bits because the switch gate capacitance increases with the significance of bits.

d. Power consumption estimation

There are DC and dynamic power consumptions. The dynamic portion originates from charging/discharging the load capacitance and the transient leakage current from the supply to the ground in the switch driver. Both are code dependent. Assuming equal probability for each code, the dynamic power related to C_L is

$$P_{load} = \sum_{n=0}^{2^N-1} C_L V_n^2 \cdot f_{clk} \cdot \frac{1}{2^N} \qquad \text{(e3.34)}$$

where C_L: load capacitance, f_{clk}: clock frequency, V_n: DAC output levels, N: number of bits. The power consumption for clock drivers is estimated in the simulation. However, the DC portion is not code dependent since the currents always flow.

$$P_{DC} = \sum_{n=0}^{N} I_n \cdot V_{DD} \qquad \text{(e3.35)}$$

where I_n: the current at the n^{th} bit, N: the number of bits, V_{DD}: the supply voltage.

e. Reproducibility discussion

The reproducibility limiting factors are memory effect, digital signal jitter and digital feedthrough to the output. All the factors influence the reproducibility as the way explained in section 3.4. The effect of the switching signal uncertainty on the DAC performance can be seen in Fig. 3.28.

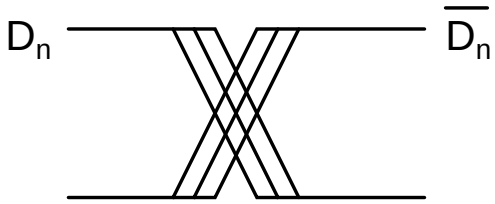


Fig. 3.28 Timing uncertainties in digital switching signals

The crossing point fluctuates due to the jitter in the digital switching signal. If the crossing point is too low, the current source may go into the linear region or even off region. The outcome is that the DAC charges the parasitic capacitance at the current source in addition to the load capacitance and it takes longer time for the DAC to settle within the required accuracy. Its effect is equivalent to delaying the switching signals. Below is the design equation to satisfy both the speed and reproducibility requirements.

$$t_{settling} > 6\sqrt{2}\sigma_{clk} + t_{r\_\max} + \beta \cdot R_L \cdot C_{L\_tot} \qquad \text{(e3.36)}$$

where $t_{settling}$: allowed settling time (500ps), σ_{clk}: clock transition jitter (5ps), t_{r_max}: maximium rise time of digital control signal to turn on the switch (often at the highest significant of bit due to large switch size), C_{L_tot}: total output capacitance, R_L: load resistance, β: constant related to the accuracy, the magnitude of digital feedthrough and component mismatch. β is ideally nine times the nominal time constant according to table 3.3. Ten can be chosen in the design to allow some room for the stochastic error and digital feedthrough (refer to section 3.4.3).

f. Simulation data

A 3-bit current steering DAC is designed and simulated in cadence to meet the design specifications. The random binary code is generated by a verilog-A script. 50Ω load is chosen in the design so that the $R_LC_{L\_tot}$ settling time satisfies the required speed and accuracy, and the biasing currents at different bits are computed to achieve 450mV output swing. 1V supply is used for digital circuit and 1.2V for analog circuit. The simulation data are first collected at the end of one clock cycle, t_s, and computed for the largest difference as the reproducible error. Then the multiple simulations are redone at different Δt ahead of t_s until the reproducible error reaches the requirement in the project. The jitter requirement is calculated from Δt according to eqn. 3.12.

Below are the design values for unit current source and switch dimension at LSB. Due to the binary weights, W/L ratio of current sources and switches at higher bits grows proportionally with theirs weights.

LSB current source W/L (μm), I_{LSB} (mA)	50/0.4, 1.3
LSB switch W/L (μm)	120/0.06
R_L (Ω)	50

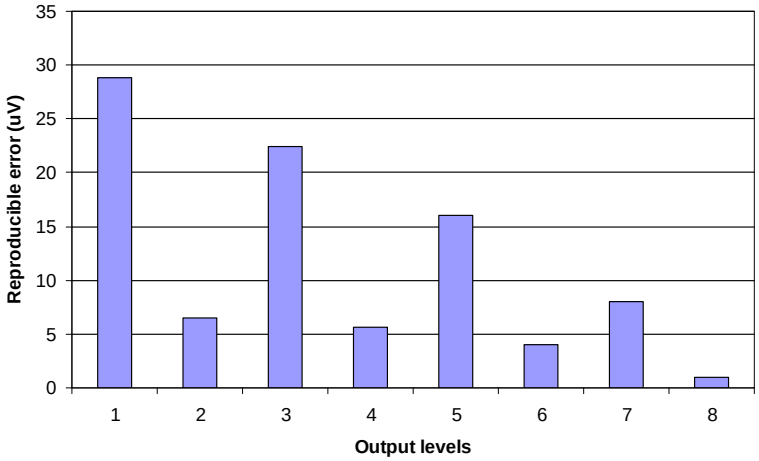


Fig. 3.29 Reproducible errors @ end of clock cycle versus different output levels

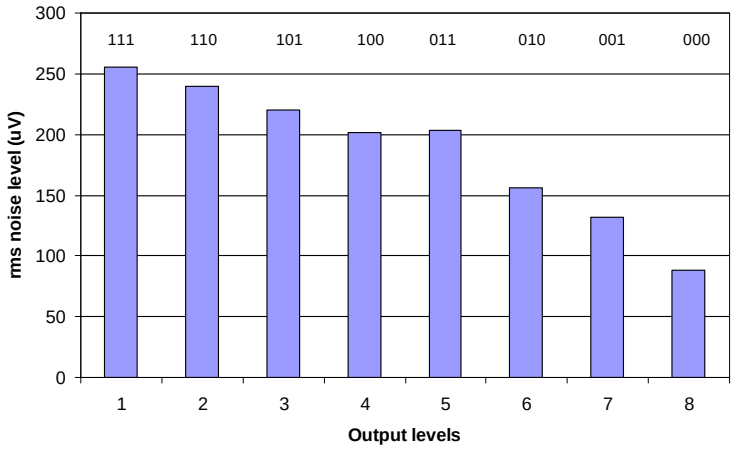


Fig. 3.29 rms noise voltage at different output levels

Below show the simulated and calculated performances.

	Δt /jitter requirement (ps)	rms noise (μV)	Total power consumption (mW)
Simulation	55/6.5	275	13.8
Calculation	315/37	131	10.8

Note: calculated total power does not include the power from switch drivers. Digital signal delay of 26ps is included in Δt estimation and β is 9.0 because full scale is 450mV and there is about 20% digital feedthrough in this design and no consideration of component spread.

Below is the decomposition of total power consumption.

	Total power consumption (mW)	DC power consumption (mW)	Dynamic power from switch drivers (mW)
simulation	13.8	10.8 (78%)	3 (12%)
calculation	10.8	10.8	x

The reproducible error increases with codes as the trend because the digital feedthrough becomes larger when there are more '1's at MSBs. Output noise also grows with codes from 88μV (about KT/C noise) to 255μV (due to more noise current

from current sources and switches), as expected. DC power consumes about eighty percent of total power, and the switch driver accounts for another 20% due to large switch capacitance to drive. The ideal Δt (315ps) deviates from the simulation data (55ps) by 260ps. This is due to clock feedthrough from the latch drivers to the output and the disturbed output has to resettle within half clock cycle again. It is shown in the figure below.

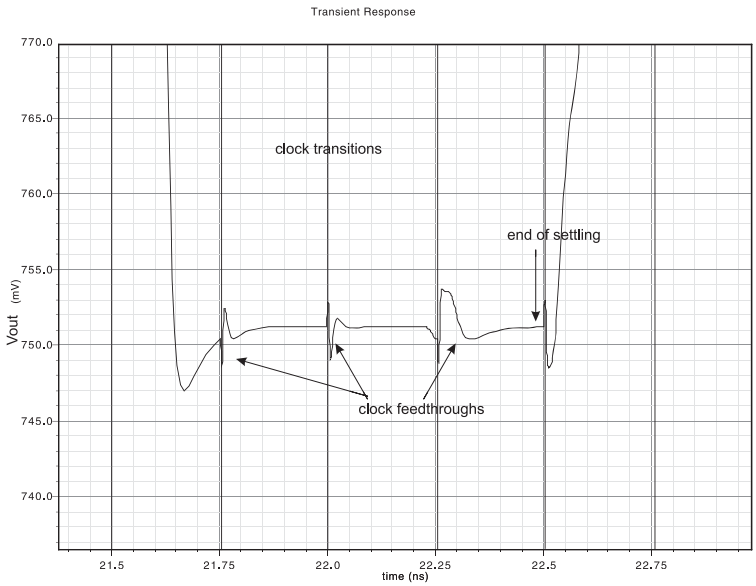


Fig. 3.30 Clock feedthrough at the output

To alleviate this effect on the output, either reducing switch size or using clock cycle less than 50% will help. The simulation is redone with 10% clock cycle, Δt improves to 102ps. The reproducible errors increase with codes, exactly as what is expected.

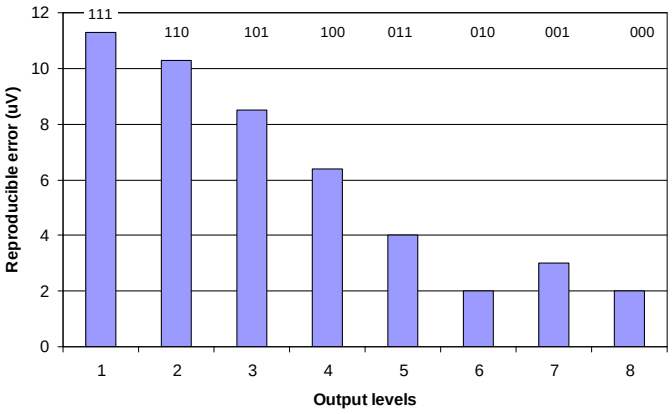


Fig. 3.31 Reproducible errors @ end of clock cycle with 10% clock cycle

3.7 Summary

First, the function of internal DAC in the two-step subranging like ADC is studied. And its specifications are derived: the internal DAC output levels are required to have

reproducible error less than 13-bit resolution and the updating rate is 2GS/s. The DAC output offset is allowed due to the calibration technique in the system.

Some common speed and reproducibility limiting factors are identified. RC settling is the fundamental speed limitation. However, there are several reproducibility limiting factors. The table below shows the contributions to the reproducible error from each source.

Error source	**Reproducible error**	**Description**
Memory effect	Maximally $V_{FS}e^{-\frac{t_s}{RC}}$	Depending on the previous code.
Jitter	Maximally $V_{FS}e^{-\frac{t_s-3\sqrt{2}\sigma_{clk}}{RC}}$	Effective settling time is shortened.
Digital/clock feedthrough	$\Delta V_{thru}e^{-\frac{t_s}{RC}}$	Modifying the initial condition
Switch control signals	$V_{FS}e^{-\frac{t_s-t_{r/f}}{RC}}$	Switch V_{gate} needs to be charged to a certain value to turn on the switch.
Component spread	$V_{FS}e^{-\frac{t_s}{(R+3\sigma_R)(C+3\sigma_C)}}$	Changing the time constant
Output noise	$6\sigma_{out\_noise}$	Random noise
Total	$(V_{FS}+\Delta V_{thru})e^{-\frac{t_s-3\sqrt{2}\sigma_{clk}-\alpha\cdot t_{r/f}}{(R+3\sigma_R)(C+3\sigma_C)}}+6\sigma_{out\_noise}$	

Note: t_s is the sampling moment. ΔV_{thru} is the feedthrough magnitude at the output. σ_{out_noise} is the standard deviation of output noise.

The reproducible error is reduced by choosing a small time constant and connecting to a large load capacitor. A general design equation is given for the required speed (specified $t_{settling}$), reproducibility (accuracy for reproducibility) and jitter requirement (σ_{jitter}). For simplicity, it is assumed that all the digital signals are fully correlated and the jitter at DAC updating moments are uncorrelated to the jitter at the sampling moments.

$$(V_{FS}+\Delta V_{thru})e^{-\frac{t_s-3\sqrt{2}\sigma_{clk}-\alpha\cdot t_{r/f}}{RC}}+6\sigma_{out\_noise}<V_{LSB@13\_bit}$$

$t_{settling}$ is 500ps in the design. α is the number of delay for the start of the settling normalized to rise or fall time. For the design equation, the optimum sampling time is at $500\,ps-3\sqrt{2}\sigma_{clk}$.

Before examing different internal DAC topologies, some design considerations are addressed. One key issue is how to measure the reproducibility for this application specific DAC. The chapter introduces a way to measure this performance. With random input codes, data are collected at the end of settling and computed for maximum reproducible error.

Afterwards, each topology is studied including choice of switches and component values, power consumption for the core part, jitter requirement and output range adjustment. Several findings are made.

- The maximum settling time of resistor string DAC is determined by the signal path in the middle.
- DC power of R-2R ladder is code dependent.
- The maximum settling time of capacitor DAC is determined by the MSB settling.
- Switches in current steering DAC are biased in saturation to reduce its parasitic capacitance contribution to the output pole. To improve reproducibility, duty cycle less than 50% is suggested to be used.

Each topology is designed for 3-bit resolution, 7.1ps clock period jitter and 100fF load capacitor, and simulated for the reproducibility at the required speed. Simulation data shows that all the topology can achieve the required specifications in the project without considering noise, but differentiate in power consumption, output noise and chip area. These differences are summarized in the following table.

Topology	Power (mW)	rms noise (μV)	Chip area (μm^2)	Overall reproducibility (bits)
Resistor string DAC	15.64	195	31.3	9
R-2R ladder DAC	1.46	225	20	9
Capacitor DAC	0.28	142	5.6×10^3	10
Current steering DAC	13.8	275	1.0×10^3	9

Note: data are achieved without the consideration of component mismatch, chip area does not include 100fF capacitor, and overall reproducibility with 3σ noise is estimated for single ended output.

Noise is the most dominant factor determining the reproducibility. With the same load (100fF), current steering DAC has the most output noise, capacitor DAC for largest chip area and highest reproducibility, resistor string DAC for largest power consumption.

Chapter 4 Internal Digital-to-Analog Converter Design in the Project

The subtraction is done in the analog domain in the two-step subranging like ADC so the analog subtraction is explained first and its specifications are derived. After that, the choice of internal DAC is made for the project to integrate itself with subtractor. The selected internal DAC is redesigned and simulated to meet the system requirements. The summary is given at last.

4.1 Analog subtraction and subtractor specifications

The differential architecture is often used due to larger signal swing, better linearity and supply voltage rejection ratio so the differential output voltage can be expressed as

$$\begin{aligned} V_{out,diff} &= A\cdot(\underbrace{V_{inp}-V_{DACp}}_{subtraction}) \underbrace{-}_{differential} A\cdot(\underbrace{V_{inn}-V_{DACn}}_{subtraction}) \\ &= A\cdot(\underbrace{V_{inp}+V_{DACn}}_{addition}) \underbrace{-}_{differential} A\cdot(\underbrace{V_{inn}+V_{DACp}}_{addition}) \end{aligned} \quad \text{(e4.1)}$$

where A: subtractor gain, V_{inp}: positive input, V_{inn}: negative input, V_{DACp}: positive DAC value, V_{DACn}: negative DAC value. The equation above indicates that there are two ways of implementing the differential output. One is taking the subtraction for input signal and DAC value of the same polarity. The other is taking the addition for input signal and DAC value of different polarity.

Below is the list of specifications that the analog subtractor needs to meet in the project.

- Input impedance: ~ 50Ω
- BW > 1GHz
  BW is with the load of hold capacitance from S/H circuit.
- THD < 13-bit resolution
- Noise: as low as possible
  If it is large, the hold capacitance from S/H circuit increases accordingly.
- Settling time to the required reproducibility < 500ps
  500ps settling time is allowed in the system because time interleaving is required due to the speed limitation of the stages after S/H circuit.

If subtracting a DAC value from the input signal, the subtractor is often active. Active subtractors hardly meet both the speed and linearity requirements. Active subtractors of high linearity (>10-bit) are often realized with OPAMP, but their speeds (less than a few hundred MHz) suffer. Active subtractors of high speed (>1GHz), such as common gate structure and one stage differential amplifier, have linearity far less than 13-bit, typically 6 – 9 bits. Therefore, the passive subtractor composed of resistors or capacitors is the choice in the project, implementing differential output by addition.

4.2 Choice of internal DAC in the project

For resistor string DAC, active subtractor is required. For current steering DAC, either buffer is required to convert voltage to current, fulfill subtraction in current domain and then convert the residue current to voltage to be sampled, or active subtractor is required to implement subtraction in voltage domain and then sampled. As discussed in the previous section, active subtractor can hardly meet both the speed and linearity requirements. It is the same to the V-I buffer. Therefore, the possible internal DACs that can integrate itself with subtraction are R-2R and capacitive internal DACs. However, the output node for capacitive internal DAC floats and static charge build-up is a concern so capacitive internal DAC is not a good choice. Below shows the suggested topology to realize the analog subtraction and DAC function.

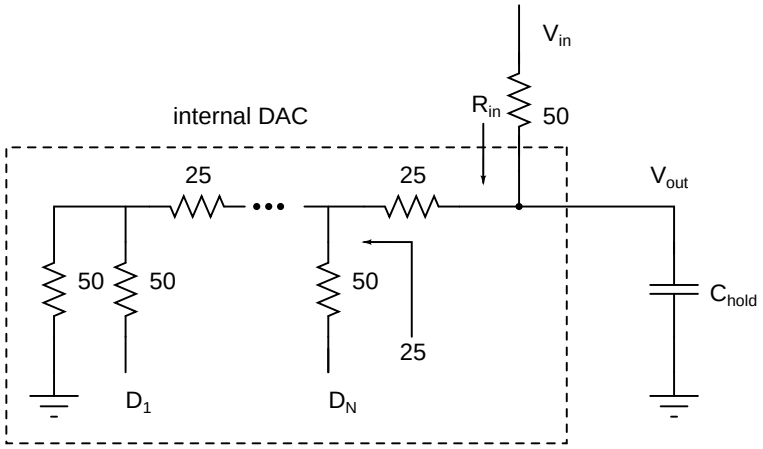


Fig. 4.1 Suggested topology for internal DAC and subtractor in the project

It is R-2R structure. R is 25Ω and 2R 50Ω for input matching. The output of one polarity (assuming positive polarity) can be expressed as

$$V_{outp} = \frac{1}{2}V_{inp} + V_{DACn,n} \text{ with n from 1 to } 2^N \qquad \text{(e4.2)}$$

where V_{outp}: output voltage of positive polarity, V_{inp}: input voltage of positive polarity, $V_{DACn,n}$: n^{th} DAC output of negative polarity. The total differential reproducible error is

$$2(V_{FS} + \Delta V_{thru})e^{-\frac{t_s - 3\sqrt{2}\sigma_{clk} - \alpha \cdot t_{r/f}}{(R+3\sigma_R)(C+3\sigma_C)}} + 6\sqrt{2}\sigma_{out\_noise} < V_{LSB@13\_bit} \qquad \text{(e4.3)}$$

Noise is the dominant factor so large capacitor helps reduce reproducible error. Given 13-bit reproducibility target (differential), the maximum allowed hold capacitance is 2pF ($\underbrace{9.7}_{14\_bit} \cdot 25\Omega \cdot C = 500\,ps$). rms noise for 2pF is 45μV so the reproducible error becomes

$$2 \cdot 61\mu V + 6 \cdot \sqrt{2} \cdot (\underbrace{45\mu V}_{thermai} + \underbrace{23\mu V}_{1/f_{noise}}) = 699\mu V$$

23μV of 1/f noise is based on the simulation for 100fF load in Chapter 3. It should be less with 2pF hold capacitance. Therefore, the achievable reproducibility for the suggested topology is between 10 and 11 bits. With other effects present in the circuit, the best realizable reproducible error is 10-bit so the following design will aim for 10-bit reproducibility.

The output range of residue signal can be defined by choosing right input range and DAC output range. Since the implementation of subtraction function is not the focus of this master assignment, only R-2R DAC is designed to meet the following specifications (single ended) with V_{FS} of 1.2V.

- 3-bit
- Speed: 2GS/s
- Reproducibility at sampling moments: <11-bit ($V_{1LSB@11\_bit}$: 488 μV)

4.3 Internal DAC design for the project

The hold capacitance of 1.5pF is chosen in the design allowing some room for other reproducible error sources. Below show the reproducible error decomposition that are aimed for in the design and at least 7.1ps clock period jitter is allowed.

Error sources	**Reproducible error**	**Value @ t_s = 475ps (μV)**
Memory effect	Maximally $V_{FS}e^{-\frac{t_s}{RC}}$	3.8
Digital signal delay	$V_{FS}e^{-\frac{t_s-t_{delay}}{RC}}$	54 (t_{delay} < 100ps)
Jitter	Maximally $V_{FS}e^{-\frac{t_s-3\sqrt{2}\sigma_{clk}}{RC}}$	6.7
Digital/clock feedthrough	$\Delta V_{thru}e^{-\frac{t_s}{RC}}$	~ 0 (low digital feedthrough)
Switch control signals	$V_{FS}e^{-\frac{t_s-t_{r/f}}{RC}}$	~ 0 (inverter R_{out} is almost constant during transitions)
Component spread	$V_{FS}e^{-\frac{t_s}{(R+3\sigma_R)(C+3\sigma_C)}}$	5.5 (σ_R/R < 1% and σ_C/C < 0.3%)
Output noise	$6\sigma_{out\_noise}$	451
Total: $(V_{FS}+\Delta V_{thru})e^{-\frac{t_s-3\sqrt{2}\sigma_{clk}-\alpha\cdot t_{r/f}}{(R+3\sigma_R)(C+3\sigma_C)}}+6\sigma_{out\_noise}$		589 (> 488μ$V_{11_bit_LSB}$)

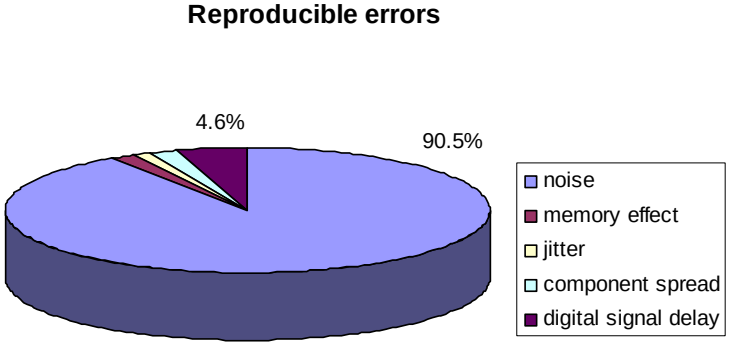


Fig. 4.2 Decomposition of reproducible errors in the design

The figure above shows that noise is the dominant reproducibility limiting factor in the design because noise improves inversely proportional to the square root of hold capacitance. The estimated output noise uses 1/f noise data for 100fF load capacitor from chapter 2 so the actual output noise should be smaller due to 15 times larger load capacitance in this design.

a . Capacitor mismatch for 1.5pF

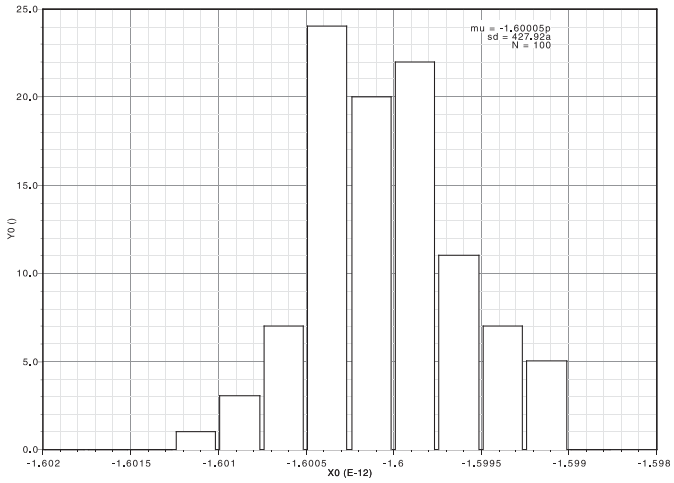


Fig. 4.3 Monte Carlo simulation for 1.5pF hold capacitance

The capacitor is metal plate type. The percentage error is 0.03%, far less than the required 0.3%.

b. Determine R, 2R and inverter sizes

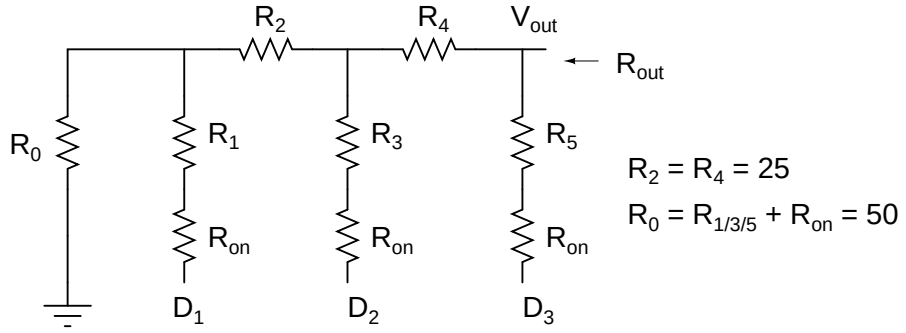


Fig. 4.4 R-2R network

In the figure above, R_{on} represents on-resistance of either PMOS or NMOS. The worst R_{on} spread is used to calculate R_{out} percentage error. Below are the chosen component values and dimensions so that $\frac{\sigma_{R_{out}}}{R_{out}} < 1\%$. The resistor is unsilicided type.

Component	Resistance (Ω)	dimension (μm)	σ (Ω)	Percentage error (%)
R_0	50.1	3.37/0.85	0.74	1.48
$R_{2/4}$	25.0	4.3/0.43	0.34	1.36
$R_{1/3/5}$	40.2	3.82/0.75	0.57	1.4
$R_{on,PMOS}$	9.5	145/0.06	0.018	0.2
$R_{on,NMOS}$	10	54/0.06	0.063	0.63

Note: table is generated with Monte Carlo simulation with N = 100.

The percentage error for R_{out} is 0.71%, less than required 1% (computed by MATLAB program, Appendix D-1).

c. Simulation data

Below are the simulated performances (Monte Carlo iteration = 10).

	rms noise (μV)	Total power consumption (mW)	Overall reproducibility @ t_s (μV)
Simulation	66.2	17.5	435 (<488)

The designed R-2R internal DAC meets the prescribed specifications: 2GS/s and 11-bit reproducibility (single ended).

4.4 Summary

This chapter briefly discusses the implementation of subtraction function. When considering integrating DAC function with analog subtraction, the choice of DAC comes only down to R-2R type. For the proposed solution, the reproducibility can not reach 13-bit because of speed requirement and the highest obtainable reproducibility is 10-bit (differential). The R-2R DAC is redesigned with 1.5pF load and the simulated performances meet the prescribed specifications.

Chapter 5 Conclusions and recommendations

5.1 Conclusions

One major contribution of this master assignment is the examination of reproducibility in different types of internal DAC, which is application specific. The research finds out different reproducibility causes present in all types of DAC, and derives a universal design equation which can be used in a specific design to meet the speed and the reproducibility requirement. Some topologies have shown that the design equation fits well to the simulation data. Simulation data has also shown that noise is the most dominant factor determining the reproducibility.

For high speed and reproducibility applications and under the same loading, current steering DAC has the most output noise, capacitor DAC for largest chip area and highest reproducibility, resistor string DAC for largest power consumption. But the design may come down to one form of output signal (voltage) and one topology (resistive) after taking into account integrating internal DAC with the subtractor. The chosen type of DAC is designed and achieves 2GS/s and 10-bit reproducibility (differential). 13-bit reproducibility can not be achieved due to the required settling speed and input matching.

In the real design, the sampling moment is better to be ahead of next DAC updating to be sure that the right sample gets sampled. Placing digital circuit away from the sensitive output node and avoiding fast signal transition during the settling time help improve reproducible error.

5.2 Recommendations

The following points may deserve further investigation.

- The effect of supply bondwire inductance on the reproducibility
- Switch on-resistance is usually nonlinear during the transitions. What on-resistance value should be used to accurately predict the reproducibility
- The effect of loading variation to the supply, for example, R-2R DAC
- Optimization to achieve low power and the required reproducibility
- Develop internal DAC design into an automation design module

APPENDIX A

1. Discussion of the system architecture with S/H circuit in the front

The conventional two-step subranging ADC has S/H circuit in the font, as shown in Fig. A.1. This section discusses the feasibility of placing S/H circuit in the front.

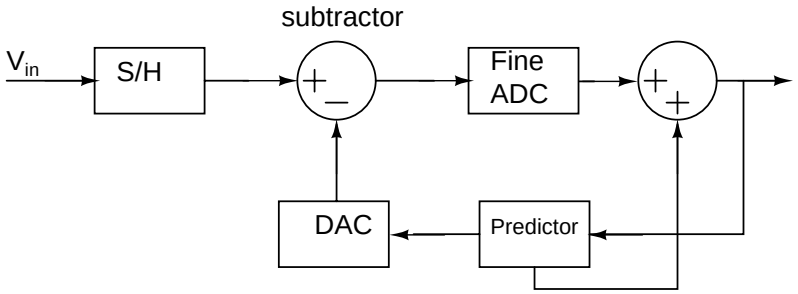


Fig. A.1 The ADC architecture with S/H circuit in the front

The input signal gets sampled and deducted by a predicted DAC value. The precision of S/H and DAC needs to be 13-bit accurate.

a. Non time interleaving solution

This solution means one S/H circuit, one DAC, one subtractor, and one fine ADC in the system. The S/H circuit tracks the input in half of clock cycle and holds the input value in the other half clock cycle so the rest of system needs to finish the conversion in half clock cycle. DAC can start the settling when the tracking starts due to the prediction in advance. Therefore, the internal DAC have less speed requirement than the rest. The first key to this solution is whether the S/H circuit can run at 2GS/s with 13-bit accuracy at large input swing. From the collected data (Table A.I), the S/H circuit hardly meets this specification. Matching can be achieved by tying 50Ω resistor at the input to the ground, providing 6.3GHz sampling bandwidth with 1pF sampling capacitance (required by the thermal noise requirement). It is far above the maximum input frequency.

b. Time interleaving with one sampling capacitor connected to the input at a time

This solution means that the S/H circuit can track and hold the input signal in 500ps with the specified accuracy, but the succeeding stages may not. The architecture and the time schedule are shown in Fig. A.2.

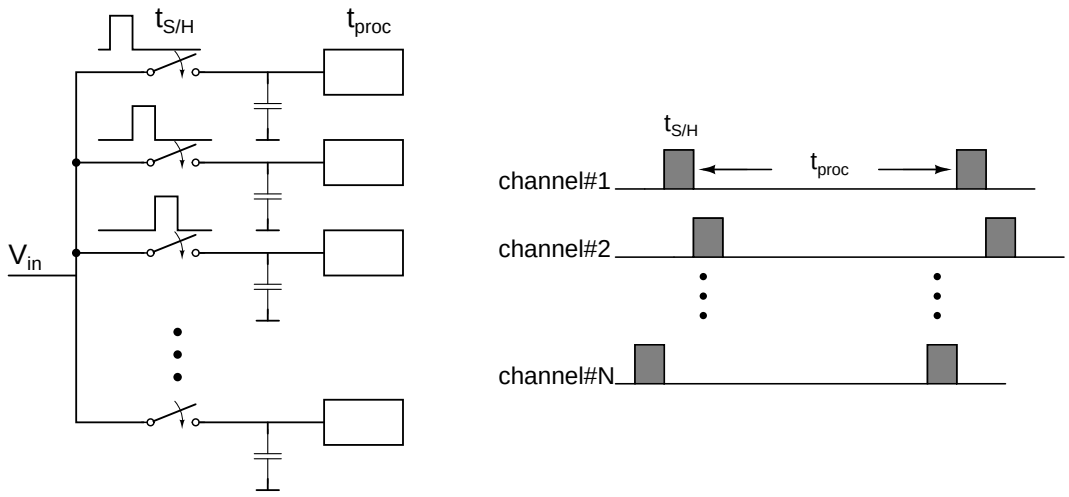


Fig. A.2 A time interleaving solution with one sampling capacitor connected to the input, and its timing diagram

In Fig. A.2, $t_{S/H}$ denotes the sampling time (one clock cycle) and t_{proc} denotes the fastest conversion time for the blocks after S/H circuit. The minimum number of time interleaving channels is calculated by $ceil\left(\frac{t_{proc}}{t_{clk@2GHz}}\right)+1$. Ceil() rounds its variable to the nearest integer toward infinity. Again, the first key question is whether S/H circuit can track and hold the input signal with 13-bit accuracy in 500ps. This solution may not work due to the same reason as in *a*.

c. Time interleaving with multiple sampling capacitors connected to the input at a time

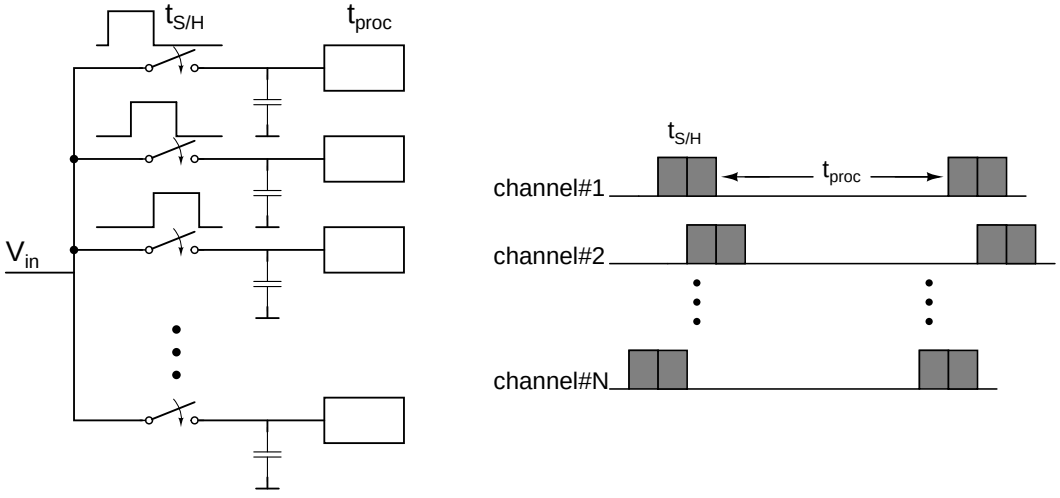


Fig. A.3 Another time interleaving solution with multiple sampling capacitors connected to the input, and its timing diagram

This solution relaxes the speed requirements on S/H circuit. Fig. A.3 shows the timing diagram for the case where two sampling capacitors connect to the input node at a time. During each sampling period, one S/H circuit is switched into the hold mode and another is switched into the track mode.

The number of sampling capacitors connected to the input node at a time is calculated by the input sampling bandwidth. Matching is done in the same way as in *a*. The sampling bandwidth is at least 1GHz (maximum input frequency), but 1.5GHz is chosen for better performance at high frequencies so the following equation needs to be satisfied.

$$\frac{1}{2\pi \cdot (50 // 50) \cdot n \cdot 1pF} > 1.5GHz$$

Thus, n < 4.2. This means that four sampling capacitors can be maximally connected to the input at the same time. The minimum number of time interleaving channels is $ceil\left(\frac{t_{proc}}{t_{clk@2GHz}}\right)+3$. The key issues are: whether the S/H circuit can track and hold the input signal with 13-bit accuracy in 2ns and what resolution time-interleaving can achieve. This solution may not work due to the same reason as in *a*.

2. Collected data from publications on S/H circuit

Table A.1 Published data on S/H circuit

Topology	Speed (MS/s)	Accuracy (bits)	Technology	Supply (V)
SH?, $2V_{pp}$ input [1]	125	14	0.35µm BiCMOS	5 and 3.3
SHA, $2V_{pp}$ input [2]	75	> 12	0.35µm CMOS	3
SH, $0.5V_{pp}$ input [3]	100	8	0.12µm CMOS	1.2

Note: SHA stands for sample and hold amplifier, SH for sample and hold circuit.

APPENDIX B

DNL and INL of both unit and binary element DACs are studied in this appendix. In the derivation, the resistor string DAC is used for the unit element DAC, and the current steering DAC with binary weights for the binary element DAC.

1. Unit element DAC

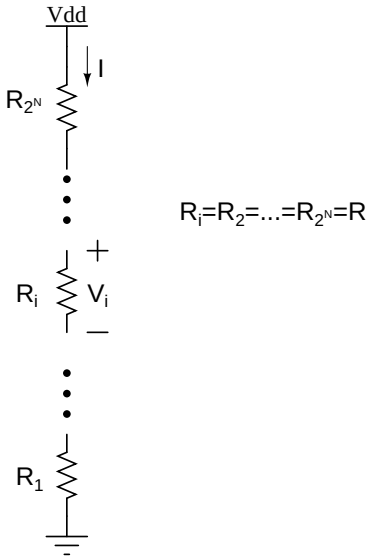


Fig. B.1 Resistor string DAC model for DNL and INL derivation

DNL and INL are calculated with regards to the average voltage step, $V_{average}$.

$$V_{average} = I \cdot R_{average} \text{ and } R = R_{average} = \frac{R_1 + R_2 + ... + R_{2^N}}{2^N}$$

Assuming that $\sigma_{R_1} = \sigma_{R_2} = ... = \sigma_{R_{2^N}} = \sigma_R$ and they are independent. DNL is expressed as

$$\begin{aligned} DNL &= \frac{I \cdot R_i - V_{average}}{V_{average}} = \frac{R_i - R_{average}}{R_{average}} \\ &= \frac{(R_{average} + 3 \cdot \sigma_R) - R_{average}}{R_{average}} = 3 \cdot \frac{\sigma_R}{R} \end{aligned} \quad \text{(B.1)}$$

The above equation shows the achievable DNL for $3\sigma_R$.

INL is expressed as

$$INL = \max\left(\frac{I \cdot \sum_{m=1}^{n} R_m - n \cdot V_{average}}{V_{average}} \right) = \frac{\max\left(\sum_{m=1}^{n} R_m \right) - n \cdot R_{average}}{R_{average}}$$

INL reaches maximum in the middle of the resistor string [7], i.e. n = N/2.

$$INL = \frac{\left(\frac{N}{2}\cdot R_{average} + 3\cdot\sqrt{2^{N-1}}\cdot\sigma_R\right) - \frac{N}{2}\cdot R_{average}}{R_{average}} = 3\cdot\sqrt{2^{N-1}}\cdot\frac{\sigma_R}{R} \qquad \text{(B.2)}$$

The above equation shows the achievable INL for $3\sigma_R$. Let INL less than 0.5, then the required resistor matching is

$$\frac{\sigma_R}{R} < \frac{1}{3\sqrt{2^{N+1}}} \qquad \text{(B.3)}$$

2. Binary element DAC

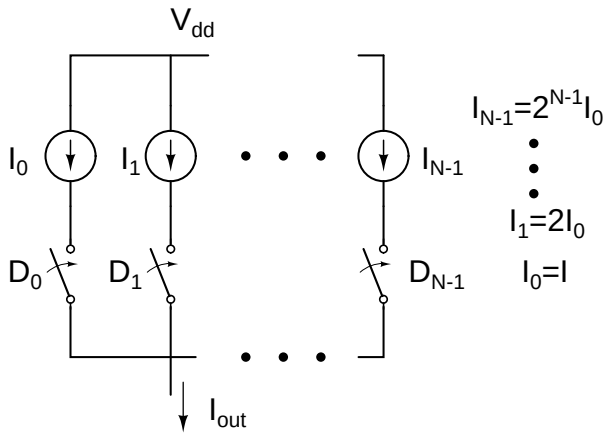


Fig. B.2 Current steering DAC model for DNL and INL derivation

Assume that the different current sources are totally uncorrelated and $\sigma_n = \sqrt{n}\sigma_I$. When the digital bits at lower significant bits are switched off and one higher significant bit is turned on, for example, the output current is switched from $(2^i - 1)\cdot I$ to $2^i\cdot I$, the standard deviation of the difference output current is big [2].

$$\sigma_{\Delta I} = \sigma_I\sqrt{2^i - 1 + 2^i} \qquad \text{(B.4)}$$

It reaches maximum when i = N – 1. Thereby,

$$DNL = 3\sqrt{2^N - 1}\,\frac{\sigma_I}{I} \qquad \text{(B.5)}$$

The above equation shows the achievable DNL for $3\sigma_I$.

The largest INL also occurs at the major transition [1] so INL is the same expression as that for unit element DAC. For the achievable INL for $3\sigma_I$ and let INL less than 0.5, the required current matching is

$$\frac{\sigma_I}{I} < \frac{1}{3\sqrt{2^{N+1}}} \qquad \text{(B.6)}$$

3. Matlab codes for R-2R DAC INL & DNL estimation

```
clear all;
clc;

% R-2R architecture, and resistor labels
%-----2R(R0)-----|
%                R (R1)
%-----2R(R2)-----|
```

```
%                R (R3)
%-----2R(R4)-----|
%                R (R5)
%-----2R(R6)-----|
%                R (R7)
%-----2R(R8)-----|
%                R (R9)
%-----2R(R10)-----|
%                R (R11)
%-----2R(R12)-----|
%                R (R13)
%-----2R(R14)-----|
%                2R (R15)
%                 |(gnd)

N=12;                                          % DAC resolution
paral = inline('a*b/(a+b)','a','b');         % the function to
calculate the total resistance in parallel

% R and 2R have the same percentage error. dR/R = d2R/2R
% Let's say 1% mismatch error and R = 1, dR = 1% and d2R = 2%
% dR is standard deviation for R values.
% d2R is standard deviation for 2R values.

for l=1:150                                              % assign the
number of runs

% assign resistor values with normal distribution
sigm=0.002;                          % standard deviation for R values
res(1:2:2*N-3)=1.0+sigm*randn(1,N-1);       % generate resistance
with odd numbers (R)
temp0(2:2:2*N+2)=2.0+2*sigm*randn(1,N+1);   % generate resistance
with even numbers (2R)
R0=temp0(2);                                % get a 2R value for R0
res(2*N-1)=temp0(2*N+2);                    % get a 2R value for R15
res(2:2:2*N-2)=temp0(4:2:2*N);

% calculate the equivalent resistance looking down at each node.
temp=res(2*N-1);
for index=2*N-1:-2:3           % index increases from bottom to up.
    temp=paral(temp,res(index-1))+res(index-2);
    reslow(N+2-(index+1)/2)=temp;
end
reslow(1)=res(2*N-1);

% calculate the equivalent resistance looking up at each node.
temp2=R0+res(1);                  % index increases from up to bottom
for index=1:2:2*N-5
    temp2=paral(temp2,res(index+1));
    resup1((index+1)/2)=temp2;
    temp2=temp2+res(index+2);
    resup2((index+1)/2)=temp2;
end

% calculate voltage divisions
div(1)=R0/(R0+res(1));
for k=2:N-1
    div(k)=resup1(k-1)/resup2(k-1);
end

ratio(N)=reslow(N)/(R0+reslow(N));
```

```
temp2=R0+res(1);
for k=N-1:-1:1
    if k==N-1
        temp=paral(reslow(k),temp2);
    else
        temp=paral(reslow(k),resup2(N-k-1));
    end
    ratio(k)=temp/(res(2*N-2*k)+temp);
end

% calculate each weight
weight(N)=ratio(N);
for k=N-1:-1:1
    weight(k)=ratio(k);
    for l=1:N-k
        weight(k)=weight(k)*div(l);
    end
end
LSB=sum(weight)/(2^N-1);

% calculate DAC outputs
for m=0:2^N-1
    temp=0;
    for k=1:N
        temp=temp+bitget(m,k)*weight(k);
    end
    Dout(m+1)=temp;
end

for m=1:2^N-1
    DNL(m)=(Dout(m+1)-Dout(m)-LSB)/LSB;
    if m>1
        INL(m)=DNL(m)+INL(m-1);
    else
        INL(1)=DNL(1);
    end
end

figure(3); plot(DNL); hold on;
figure(4); plot(INL); hold on;
end
```

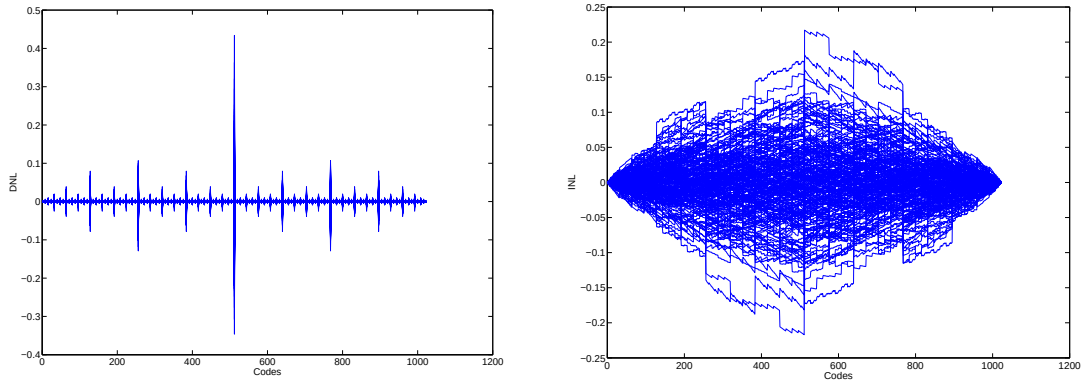


Fig. B.3 DNL and INL plots with 200 runs for N = 10 and 0.2% mismatch

The plots above demonstrate the achievable DNL and INL with the specified resistor mismatch, and also prove that both DNL and INL is the worst at the major transition.

APPENDIX C

1. DAC reproducibility requirement

The requirement is that SNR at the sampling moments is targeted to be 13-bit resolution. This means that the total noise and nonlinearity, contributed by the subtractor, the internal DAC and S/H circuit, at the sampling moments should be less than 13-bit resolution. Both subtractor and S/H circuit are assumed to be linear and noiseless in the reproducibility derivation since the focus of the thesis is the internal DAC design. In order to obtain the requirement for the internal DAC, the subtractor gain needs to be defined. The gain is less than unity due to the following reasons.

1. Subtractor with OPAMP has high linearity, but the bandwidth can hardly reach more than 1GHz.
2. Active subtractor like differential amplifier with active load is preferred to have unity gain to achieve large signal bandwidth. (active load is necessary to convert differential input to single ended output.)
3. Passive subtractor can achieve high speed and linearity, but the gain is maximally unity.

Therefore, the reproducibility requirement on the internal DAC is 13-bit.

2. MATLAB code for system simulation

```
clear all;
clc;

% the program is for 2 bits for the first stage, and 2 bits for the
second
% stage
N = 2;                                          % number of bits for
the first stage
L = 2;                                          % number of bits for
the second stage
input = 0:0.0001:1;                             % generate input
signal

% assign resistor values with normal distribution
sigm=0.08;                                      % standard deviation
for R values
res(1:2^N)=1.0+sigm*randn(1,2^N);
dac(1)=0;                                       % assign the first
DAC value
dac(2^N+1)=1;                                   % assign the last DAC
value
tot=sum(res(1:end));

% calculate the actual dac levels
for n=2:2^N                                     % generate dac output
levels from resistor string with mismatches
    dac(n)=sum(res(1:n-1))/tot;                 % the actual output
levels
end

% calculate the ideal levels
```

```
for n=1:2^N+1                                   % generate ideal
output levels
    idac(n)=(n-1)*1/2^N;
end

M = length(input);                              % get the length of
input signal
level(1:M)=0;
code1(1:M)=0;                                   % the coarse code
for n=1:M
    sgn=0;
    p=1;
    while (sgn==0)                              % find the right
range for each input signal amplitude
        if ((input(n)<idac(p+1))&&(input(n)>=idac(p)))
            sgn = 1;
            level(n) = dac(p);                  % the actual dac
level which needs to be subtracted by input(n)
            code1(n) = (p-1)*2^N;
        end
        if (input(n)==1.0)
            sgn = 1;
            level(n) = dac(2^N);
            code1(n) = (2^N-1)*2^N;
        end
        p=p+1;
    end
    resdue(n)=2^N*(input(n)-level(n));
end

% number of bits for the ideal second stage
dac2=0:1/2^L:1;                                 % dac2 are ideal
levels for the second stage.
for n=1:M
    sgn=0;
    p=1;
    while (sgn==0)
        if (resdue(n)>=1.0)                     % the situation where
residue > upper limit
            sgn=1;
            code2(n)=2^L-1;
        end
        if (resdue(n)<=0.0)                     % the situation where
residue < lower limit
            sgn=1;
            code2(n)=0;
        end
        if ((resdue(n)<dac2(p+1))&&(resdue(n)>=dac2(p))&&(sgn==0))
            sgn=1;                              % the situation where
residue is in the range
            code2(n)=p-1;
        end

        p=p+1;
    end
end

code = code1+code2;
plot(input,code,'--r');
hold on;
```

```
set(gca,'fontsize',24);
ylabel('Codes','fontsize',24);
xlabel('Input','fontsize',24);
```

3. Required settling time for stochastic error

The dynamic settling error is

$$error = V_s e^{-\frac{t}{RC}} \tag{C.1}$$

$$\begin{aligned}\sigma_{error}^2 &= \left(\frac{\partial error}{\partial R}\right)^2 \sigma_R^2 + \left(\frac{\partial error}{\partial C}\right)^2 \sigma_C^2 \\ &= \left(\frac{V_s t}{RC} e^{-\frac{t}{RC}}\right)^2 \left(\frac{\sigma_R}{R}\right)^2 + \left(\frac{V_s t}{RC} e^{-\frac{t}{RC}}\right)^2 \left(\frac{\sigma_C}{C}\right)^2 \\ &= error^2 \cdot \left(\frac{t}{RC}\right)^2 \left(\frac{\sigma_R}{R}\right)^2 + error^2 \cdot \left(\frac{t}{RC}\right)^2 \left(\frac{\sigma_C}{C}\right)^2\end{aligned}$$

Let

$$\begin{aligned}&error + 3\sigma_{error} < V_{LSB} \\ &error \cdot \left[1 + \frac{3t}{RC} \cdot \sqrt{\left(\frac{\sigma_R}{R}\right)^2 + \left(\frac{\sigma_C}{C}\right)^2}\right] < V_{LSB} \\ &e^{-\frac{t}{RC}} \cdot \left[1 + \frac{3t}{RC} \cdot \sqrt{\left(\frac{\sigma_R}{R}\right)^2 + \left(\frac{\sigma_C}{C}\right)^2}\right] < 2^{-N}\end{aligned} \tag{C.2}$$

4. Random walking one code generator

The following code generator is for 3 bits. Four-bit generator can be constructed similarly.

```
// veriloga for random_walk1_3bitcode
`include "constants.h"
`include "disciplines.h"

// D0 D1 D2 D3 are inverted.
// D0 D1 D2 D3: 0 1 1 1  if the random number is 4N+1;
// D0 D1 D2 D3: 1 0 1 1  if the random number is 4N+2;
// D0 D1 D2 D3: 1 1 0 1  if the random number is 4N+3;
// D0 D1 D2 D3: 1 1 1 0  if the random number is 4N;
// digital signal lasts for one period.

module random_walk1_1bitcode (vout7, vout6, vout5, vout4, vout3,
vout2, vout1, vout0);
output vout7, vout6, vout5, vout4, vout3, vout2, vout1, vout0;
electrical vout7, vout6, vout5, vout4, vout3, vout2, vout1, vout0;
parameter integer seed = 92;
parameter real tperiod = 1 from (0:inf);
parameter real vlogic_high = 1.2;
parameter real vlogic_low  = 0 ;
parameter real tdel=0 from [0:inf);
parameter real trise=1p;
parameter real tfall=1p;

   real next;
```

```
integer b7, b6, b5, b4, b3, b2, b1, b0;

real vout0_val, vout1_val, vout2_val, vout3_val;
real vout4_val, vout5_val, vout6_val, vout7_val;
integer iseed;
integer bitreg;

analog begin

@ (initial_step) begin
   iseed = seed;
   next = $abstime + tperiod;
   vout0_val = vlogic_low;      vout1_val = vlogic_low;
   vout2_val = vlogic_low;      vout3_val = vlogic_low;
   vout4_val = vlogic_low;      vout5_val = vlogic_low;
   vout6_val = vlogic_low;      vout7_val = vlogic_low;
end

bitreg = {$random(iseed)}%8;
@ ( timer(next) ) begin

   if ( bitreg == 0 ) begin
    b0 = bitreg & 00;
    b1 = (bitreg+1) & 01;   b2 = (bitreg+1) & 01;
    b3 = (bitreg+1) & 01;   b4 = (bitreg+1) & 01;
    b5 = (bitreg+1) & 01;   b6 = (bitreg+1) & 01;
    b7 = (bitreg+1) & 01;
   end

   if ( bitreg == 1 ) begin
      b0 = bitreg & 01;
    b1 = bitreg & 00;       b2 = bitreg & 01;
    b3 = bitreg & 01;       b4 = bitreg & 01;
    b5 = bitreg & 01;       b6 = bitreg & 01;
    b7 = bitreg & 01;
   end

   if ( bitreg == 2 ) begin
   b0 = (bitreg+1) & 01;    b1 = (bitreg+1) & 01;
    b2 = bitreg & 00;       b3 = (bitreg+1) & 01;
    b4 = (bitreg+1) & 01;   b5 = (bitreg+1) & 01;
    b6 = (bitreg+1) & 01;   b7 = (bitreg+1) & 01;
   end

   if ( bitreg == 3 ) begin
      b0 = bitreg & 01;     b1 = bitreg & 01;
    b2 = bitreg & 01;       b3 = bitreg & 00;
    b4 = bitreg & 01;       b5 = bitreg & 01;
    b6 = bitreg & 01;       b7 = bitreg & 01;
   end

   if ( bitreg == 4 ) begin
      b0 = (bitreg+1) & 01;       b1 = (bitreg+1) & 01;
    b2 = (bitreg+1) & 01;   b3 = (bitreg+1) & 01;
    b4 = bitreg & 00;       b5 = (bitreg+1) & 01;
    b6 = (bitreg+1) & 01;   b7 = (bitreg+1) & 01;
   end

   if ( bitreg == 5 ) begin
      b0 = bitreg & 01;     b1 = bitreg & 01;
    b2 = bitreg & 01;       b3 = bitreg & 01;
```

```
       b4 = bitreg & 01;       b5 = bitreg & 00;
       b6 = bitreg & 01;       b7 = bitreg & 01;
      end

      if ( bitreg == 6 ) begin
         b0 = (bitreg+1) & 01;       b1 = (bitreg+1) & 01;
       b2 = (bitreg+1) & 01;   b3 = (bitreg+1) & 01;
       b4 = (bitreg+1) & 01;   b5 = (bitreg+1) & 01;
       b6 = bitreg & 00;       b7 = (bitreg+1) & 01;
      end

      if ( bitreg == 7 ) begin
         b0 = bitreg & 01;     b1 = bitreg & 01;
       b2 = bitreg & 01;       b3 = bitreg & 01;
       b4 = bitreg & 01;       b5 = bitreg & 01;
       b6 = bitreg & 01;       b7 = bitreg & 00;
      end

         vout0_val = (vlogic_high-vlogic_low)*b0 + vlogic_low;
         vout1_val = (vlogic_high-vlogic_low)*b1 + vlogic_low;
         vout2_val = (vlogic_high-vlogic_low)*b2 + vlogic_low;
         vout3_val = (vlogic_high-vlogic_low)*b3 + vlogic_low;
         vout4_val = (vlogic_high-vlogic_low)*b4 + vlogic_low;
         vout5_val = (vlogic_high-vlogic_low)*b5 + vlogic_low;
         vout6_val = (vlogic_high-vlogic_low)*b6 + vlogic_low;
         vout7_val = (vlogic_high-vlogic_low)*b7 + vlogic_low;
         next = next + tperiod;
      end

      V(vout0) <+ transition(vout0_val,tdel,trise,tfall);
      V(vout1) <+ transition(vout1_val,tdel,trise,tfall);
      V(vout2) <+ transition(vout2_val,tdel,trise,tfall);
      V(vout3) <+ transition(vout3_val,tdel,trise,tfall);
      V(vout4) <+ transition(vout4_val,tdel,trise,tfall);
      V(vout5) <+ transition(vout5_val,tdel,trise,tfall);
      V(vout6) <+ transition(vout6_val,tdel,trise,tfall);
      V(vout7) <+ transition(vout7_val,tdel,trise,tfall);
    end
endmodule
```

5. Cadence script to collect data for transient response

```
ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design(
"/misc/icenas/weiz/simulation/cDAC3bit_BS/spectre/schematic/netlist/n
etlist")
resultsDir(
"/misc/icenas/weiz/simulation/cDAC3bit_BS/spectre/schematic" )
modelFile(

'("/misc/icetux6/cadappl/iclibs/CMOS065/PcCMOS065_7M4X0Y2ZPTP/2.0/lib
/CMOS065_7M4X0Y2Z/../../tools/models/LP_50A/mm11/spectre/b6/include_n
ominal.scs" "")

'("/misc/icetux6/cadappl/ictools/cadence_ic/5.1.41.ISR/tools.lnx86/df
II/etc/cdslib/artist/functional/allFunc.scs" "")
)
analysis('tran ?stop "150n"  ?errpreset "conservative"  ?maxstep
"50p"  )
desVar(       "period" 500p   )
desVar(       "delay" 0 )
```

```
temp( 27 )

out = outfile("./CapDAC/paramResults3bits2GBSjit1.out" "w")
run()
results()
selectResults('tran)
outputs()

period=0.5e-9
sample=0.26e-9
offset=20e-9
delay=25e-12
for(tt 0 300
    time = tt*period+offset+sample-delay
    fprintf(out "%1.5e  " time)
    fprintf(out "%1.6e \n" value(VT("/out"),time) )
)

close(out)
```

6. Matlab file to obtain reproducible error from the simulated data

The file names should be changed accordingly.

```
clear all;  clc;

format long;
filename = 'U:\cadence\c65\CapDAC\paramResults3bits2Gjit1.out';
fid = fopen(filename,'r');

Z = fscanf(fid,'%g %g',[2 inf])';
t = Z(:,1);     out = Z(:,2);

fclose(fid);

N = 3;                              % the number of bits
C(1)=143;                           % assign unit capacitance 67/143
Cl=100;                             % assign load capacitance
for m=2:N                           % calculate different capacitance
    C(m)=2*C(m-1);
end
Ct = sum(C(1:end));                 % calculate the total capacitance
LSB = 1.2/2^N;                      % ideal LSB
dac(1:2^N)=0;
for m=0:2^N-1                       % calculate actual dac levels
    for l=1:N
        dac(m+1)=bitget(m,l)*C(l)/(Ct+Cl)+dac(m+1);
    end
    dac(m+1)=1.2*dac(m+1);
end

len = length(out);    num(1:2^N) = 0;
error(1:2^N,:) = 0;   store_cyl(1:2^N) = 0;
dac_data(1:2^N) = 0;
for index = 1:len
    cyl = 1;    sgn = 1;
    while sgn == 1
        if ( abs(out(index)-dac(cyl)) < 0.8*LSB )
            if (store_cyl(cyl)==0)
                store_cyl(cyl)=cyl;
                dac_data(cyl)=out(index);
            end
```

```
            sgn = 0;
            %temp = size(error(cyl,:),2);
            num(cyl)=num(cyl)+1;
            output_level(cyl,num(cyl)) = out(index);            %
find which level
            output_index(cyl,num(cyl)) = index;
        end
        cyl=cyl+1;
    end
end

len = length(output_level(1,:));
for cyl = 1:8
    minimum(cyl)=output_level(cyl,1);
    for index =2:len
        if (output_level(cyl,index)>0 &&
output_level(cyl,index)<minimum(cyl))
            minimum(cyl)=output_level(cyl,index);
        end
        if (output_level(cyl,index)>0)
            end_index(cyl)=index;
        end
    end
end

for cyl = 1:8
    figure(cyl);
    error=output_level(cyl,1:end_index(cyl))-minimum(cyl);
    stem(error*1e6);
    set(gca,'fontsize',16);
    xlabel('samples','fontsize',16);
    ylabel('error for the first level (uV)','fontsize',16);
end
```

7. Equivalent resistance in R-2R ladder DAC

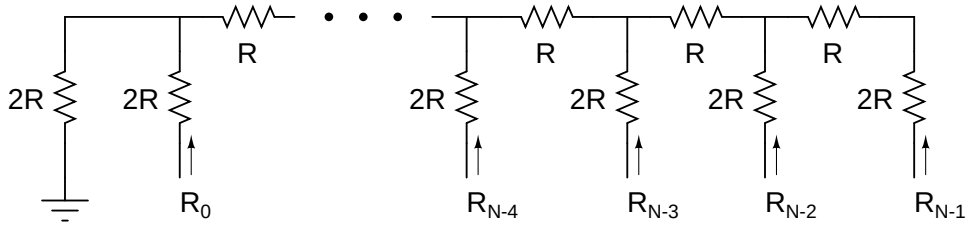


Fig. C.1 R-2R ladder

It can be derived that

$$R_{N-1} = 2R + 2R,\ R_{N-1} = 2R + \frac{6}{5}R,$$

$$R_{N-1} = 2R + \frac{22}{21}R,\ R_{N-2} = 2R + \frac{86}{85}R, \ldots$$

Look at the denominator in the second term.

1 = 1

5 = 1 + 4

21 = 1 + 4 + 16

85 = 1 + 4 + 16 + 64

…

Therefore,

$$deno\min ator = \frac{4^{N-n}-1}{3} \quad \text{with n = 0, 1, …, N-1}$$

$$R_n = 2R + \frac{4^{N-n}+2}{4^{N-n}-1}R \quad \text{with n = 0, 1, …, N-1} \qquad \text{(C.3)}$$

8. R-2R power coefficients

```
% program to estimate DC and dynamic power dissipation for R-2R
architecture
clear all; clc;

% R-2R architecture, and resistor labels
%-----2R(R0)-----|
%                R (R1)
%-----2R(R2)-----|
%                R (R3)
%-----2R(R4)-----|
%                R (R5)
%-----2R(R6)-----|
%                R (R7)
%-----2R(R8)-----|
%                R (R9)
%-----2R(R10)-----|
%                R (R11)
%-----2R(R12)-----|
%                R (R13)
%-----2R(R14)-----|
%                2R (R15)
%                 |(gnd)

N=10;                                    % resolution
paral = inline('a*b/(a+b)','a','b');

% R and 2R are normalized to 1 and 2 in the program.
% assign resistor values, 1 or 2.
res(1:2:2*N-3)=1.0;        % generate resistance with odd numbers (R)
temp0(2:2:2*N+2)=2.0;      % generate resistance with even numbers
(2R)
R0=temp0(2);               % get a 2R value for R(0)
res(2*N-1)=temp0(2*N+2);   % get a 2R value for R(2^N)-1
res(2:2:2*N-2)=temp0(4:2:2*N);

% calculate the equivalent resistance looking down at each node.
temp=res(2*N-1);
for index=2*N-1:-2:3       % index increases from bottom to up.
    temp=paral(temp,res(index-1))+res(index-2);
    reslow(N+2-(index+1)/2)=temp;
end
reslow(1)=res(2*N-1);

% calculate the equivalent resistance looking up at each node.
temp2=R0+res(1);           % index increases from up to bottom
for index=1:2:2*N-5
    temp2=paral(temp2,res(index+1));
    resup1((index+1)/2)=temp2;
    temp2=temp2+res(index+2);
    resup2((index+1)/2)=temp2;
```

```
end

% calculate the equivalent resistance looking into each branch.
Rb(N) = 2+reslow(N);
Rb(N-1) = 2+paral(3,reslow(N-1));
if N>2
    for k=N-2:-1:1
        Rb(k) = 2+paral(reslow(k),resup2(N-k-1));
    end
end

% calculate the power coefficients
arafa=0;                    % DC power coefficient
Reqn=0;                     % temporary variable
for m=0:2^N-1
    sgn=0;
    Reqn=inf;
    for k=1:N
        temp=0;
        temp=bitget(m,k);   % get the bit from low to high
        if ( (temp==1) && (sgn==0) )   % if bit is 1 and first 1, do
the following
            Reqn=Rb(k);
            sgn=1;
        else if ( (temp==1) && (sgn==1) )    % if there are more than
two 1s, do the following
            Reqn=paral(Reqn,Rb(k));
            end
        end
    end
    arafa=arafa+1/2^N/Reqn;
end

beta=0;                     % dynamic power coefficient
for m=0:2^N-1
    beta=beta+m^2/2^(3*N);
end
```

9. Capacitor DAC settling

$$C_{n,eq}^{'} = \frac{(C_L + C_t - C_n)\cdot C_n}{C_t + C_L} \tag{C.4}$$

where C_L: load capacitance, C_t: the total capacitance, $\sum_{n=1}^{N} C_n$. The elementary capacitors are in binary weights. C_L and C_t are constants. If $C_{n+1,eq}^{'} - C_{n,eq}^{'} > 0$, then $C_{n,eq}^{'}$ increases with n and reaches maximum at N.

$$\frac{(C_L + C_t - C_{n+1})\cdot C_{n+1}}{C_t + C_L} = \frac{(C_L + C_t - 2C_n)\cdot 2C_n}{C_t + C_L}$$

Then

$$C_{n+1,eq}^{'} - C_{n,eq}^{'} = \frac{(C_L + C_t - 3C_n)\cdot C_n}{C_t + C_L}$$

Because $C_t - 3C_n = C_1 + ... + \underbrace{C_n + 2C_n}_{=3C_n} + ... + C_N - 3C_n > 0$, then $C_{n+1,eq}^{'} - C_{n,eq}^{'} > 0$.

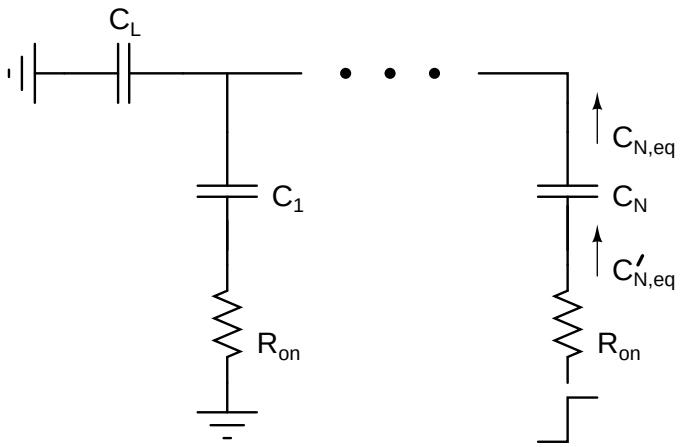


Fig. C.2 Capacitor DAC settling model

R_{on} in the figure designates PMOS or NMOS switch on-resistance at different bits and they are equal. The maximum output reproducible error occurs when the output settles from 11…1 or 00…0 to 10…0 because the RC time constant is largest at the MSB. To meet 13-bit accuracy,

$$t_{settling} > 6\sqrt{2}\sigma_{clk} + \beta \cdot \alpha \cdot R_{on} \cdot C'_{N,eq} \qquad \text{(C.5)}$$

where $t_{settling}$: the allowed settling time (500ps), σ_{clk}: clock transition jitter (5ps), R_{on}: the switch on-resistance, β: constant related to the accuracy and component mismatch, α: scaling factor related to R_{on} and resolution. α is between $1+\frac{1}{2^{N-1}-1}$ and 2.

The largest RC time constant is about $2 \cdot R_{on} \cdot C'_{N,eq}$, as shown in the figure below.

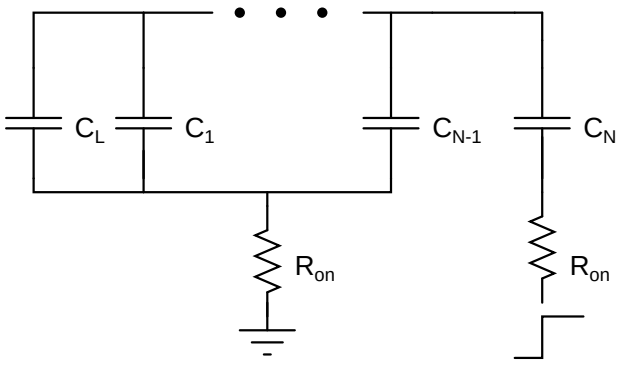


Fig. C-3 Up bound estimation for the time constant

All the capacitance except C_N adds up and R_{on} is not attenuated by the parallel structure. The simulation proves this conclusion by sweeping R_{on} from 20Ω to 1kΩ with C_t = 1pF.

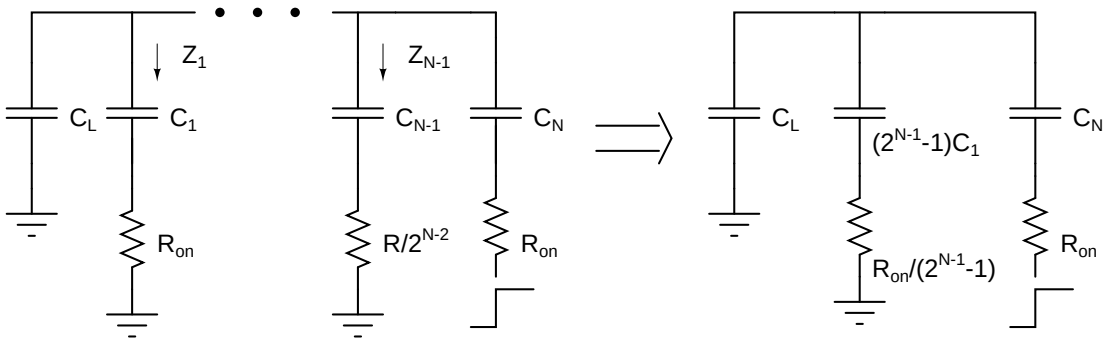


Fig. C-4 Lower bound estimation for the time constant

For lower bound estimation, the switch on-resistance $\left(\frac{R_{on}}{2^{n-2}}\right)$ is reduced proportionally to the binary weight so that RC product for each branch is constant ($C_n = 2^{n-1}C_1$ and $R_{on,n} = R_{on}/2^{n-1}$). The time constant for this case is smaller than the case where all the switch on-resistances are kept the same as R_{on}.

$$Z_n = \frac{1}{2^{n-1}}\left(R_{on} + \frac{1}{sC_1}\right)$$

so that

$$Z_1 // Z_2 // ... // Z_{N-1} = \frac{1}{2^{N-1}-1}\left(R_{on} + \frac{1}{sC_1}\right)$$

The equivalent on-resistance is $\left(1 + \frac{1}{2^{N-1}-1}\right)R_{on}$.

10. Matlab codes for power consumption of capacitor DAC

```
clear all;clc;

N = 3;                                  % number of bits
vdd = 1.2;                              % supply voltage
fclk = 2e9;                             % clock frequency
Ct = 1e-12;                             % total capacitance
C(1)=1e-12/(2^N-1);                     % calculate unit capacitance in
fF
Cl=100e-15;                             % load capacitance, 100fF
for m=2:N                               % calculate different capacitance
    C(m)=2*C(m-1);
end
dac(1:2^N)=0;
for m=0:2^N-1                           % calculate actual dac levels
    for l=1:N
        dac(m+1)=bitget(m,l)*C(l)/(Ct+Cl)+dac(m+1);
    end
    dac(m+1)=1.2*dac(m+1);
end

Eg(1:2^N)=0;                            % the energy for 2^N codes
for m = 1:2^N
    Eg(m)=Cl*dac(m)^2/2;                % energy stored in load
```

```
    for l = 1:N
        Eg(m)=Eg(m)+bitget(m,l)*C(l)*dac(m)^2/2+(1-
bitget(m,l))*C(l)*(vdd-dac(m))^2/2;
    end
end
P=2*sum(Eg)*fclk/2^N;
```

11. Output noise level of capacitive DAC

The following derivation only considers thermal noise across capacitors.

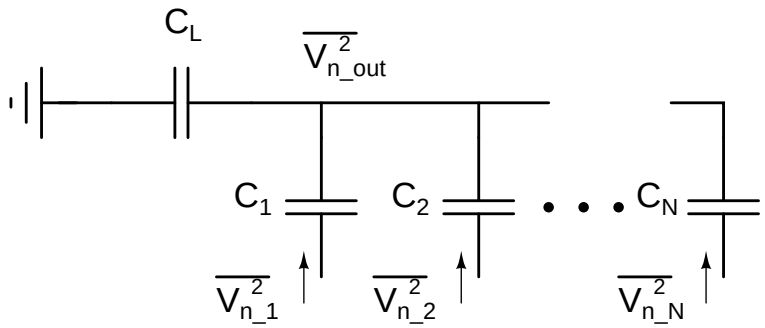


Fig. C-5 Noise model for capacitive DAC

$\overline{V_{n\_n}^2}$ with n from 1 to N represents the thermal noise variance looking into each capacitor branch. Let $C_t = \sum_{n=1}^{N} C_n$.

$$\overline{V_{n\_n}^2} = \frac{KT}{C_{n\_eq}} \text{ where } C_{n\_eq} = \frac{C_n(C_t - C_n + C_L)}{C_t + C_L}$$

So its contribution to output noise variance is

$$\overline{V_{n\_out,n}^2} = \frac{KT(C_t + C_L)}{C_n(C_t - C_n + C_L)} \cdot \frac{C_n^2}{(C_t + C_L)^2}$$

$$= \frac{KT}{(C_t + C_L)} \cdot \frac{C_n}{(C_t - C_n + C_L)}$$

Therefore, total output noise variance is

$$\overline{V_{n\_out,n}^2} = \frac{KT}{C_L + C_t} \cdot \sum_{n=1}^{N} \frac{C_n}{C_L + C_t - C_n} \qquad \text{(C.6)}$$

APPENDIX D

1. MATLAB program to compute percentage error

```
for n = 1:1000
    R0 = 50.1+50.1*0.0148*randn(1);
    R2 = 25+25*0.0136*randn(1);
    R4 = 25+29.2*0.0136*randn(1);
    R1 = 40.2+42.7*0.014*randn(1)+10+10*0.01*randn(1);
    R3 = 40.2+42.7*0.014*randn(1)+10+10*0.01*randn(1);
    R5 = 40.2+42.7*0.014*randn(1)+10+10*0.01*randn(1);

    temp = paral(R0,R1);
    temp = temp+R2;
    temp = paral(temp,R3);
    temp = temp+R4;
    temp = paral(temp,R5);
    Rout(n) = temp;
end

ave_Rout = mean(Rout);          % calculate average Rout
dev_Rout = std(Rout);           % calculate sigma of Rout
per_Rout = dev_Rout*100/ave_Rout;   % calculate percentage error
```

Bibliography

[1] Rudy J. van de Plassche: CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters, 2nd edition, Kluwer Academic Publishers, 2003

[2] Marcel Pelgrom, Analog-to-digital course handout

[3] Walt Kester: Data Conversion Handbook, ELSEVIER, 2005

[4] Razavi, Behzad: Principles of data conversion system design, IEEE Press, 1995

[5] Emil Arnold, Theodore Letavic, and Sam Herko, "High-Field Electron Velocity in Silicon Surface-Accumulation Layers", IEEE ELECTRON DEVICE LETTERS, VOL. 20, NO. 9, SEPTEMBER 1999

[6] Anne Van den Bosch, Marc A. F. Borremans, Michel S. J. Steyaert, and Willy Sansen, "A 10-bit 1-GSample/s Nyquist Current-Steering CMOS D/A Converter", IEEE Journal of Solid-State Circuits, Volume 36, No. 3, March 2001

[7] Doernberg, J., Gray, P.R., Hodges, D.A., "A 10-bit 5-Msample/s CMOS two-step flash ADC", IEEE Journal of Solid-State Circuits, Volume 24, Issue 2, April 1989 Page(s):241 - 249

[8] Ahmed M.A. Ali, Chris Dillon, Robert Sneed, Andy Morgan, John Kornblum, Lu Wu and Scott Bardsley, "A 14-bit 125 MS/s IF/RF Sampling Pipelined A/D Converter", IEEE 2005 CUSTOM INTEGRATED CIRCUITS CONFERENCE

[9] Echere Iroaga, Boris Murmann "A 12-Bit 75-MS/s Pipelined ADC Using Incomplete Settling", IEEE Journal of Solid-State Circuits, Volume 42, No. 4, April 2007 Page(s):748 – 756

[10] Louwsma, S.M., Tuijl, A.J.M. van, Vertregt, M., Scholtens, P.C.S., Nauta, B. "An Interleaving Track & Hold with 7.6 ENOB @ 1.6 GS/s in 0.12 μm CMOS", ProRISC Veldhoven, 2004, Book of abstracts: Page 56