Hochschule Kempten
University of Applied Sciences

Low Cost Serial DAC Simulation, Realization, Error Correction and Characterization

Jörg Vollrath

***Abstract*—A serial C DAC has a minimum number of components. This structure makes it easy to simulate, realize, characterize and to study error correction. This paper presents a discrete 12 bit serial C DAC with digital error correction. Theory, high level (JavaScript), low level (LTSPICE) simulations, a real circuit and low cost measurements and error correction using an Arduino board are presented. Comparing theory, simulation and measurement gives deeper understanding of DAC leading to better circuits and improved characterization methods. The measurements show INL, DNL below 4 and FFT with 46.5 dB SNQR which gives 6.5 ENOB at 33 Hz sampling frequency of a 12 bit DAC. Hardware costs are below 100 Euro without an oscilloscope for detailed signal measurements.**

***Index Terms*—serial DAC, circuit simulation, characterization, INL, DNL, SNR, error correction**

I. Introduction

Data converters are very important elements connecting physical world to electrical world allowing digital signal processing. The quality of digital processing depends on the performance of the data converters. Therefore understanding and measurement of the quality of data converters allow to build high performance digital systems.

The resolution of data converters is limited by offset error, gain error, INL, DNL and SNR. The IEEE standards 1057 and 1241 are applicable for data converters. For high resolution data converters a lot of effort is required to calculate and measure these numbers. Unfortunately the relationship between the circuit configuration, faults and changes in these parameters is not easily determined. In literature there are only limited data connecting circuit faults and errors to patterns in INL, DNL and SNR. A direct link between INL and spectrum is mathematically difficult. There are limited examples showing improvements of digital error correction [1, 2, 3]. Therefore publicly available tools to study these effects are very interesting.

Test and characterization engineers can use these tools to be able to prepare very early in the design process a proper test and characterization method. Simulations can be directly transferred to tests. Additional tests can be simulated before hardware is available.

Jörg Vollrath, Joerg.vollrath@hs-kempten.de. University of Applied Science, Bahnhofstraße 61, 87435 Kempten, Germany.

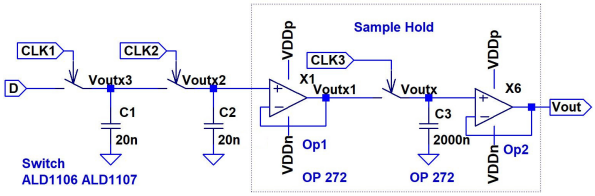

Figure 1. A serial DAC with sample and hold.

Test data analysis can be prepared using available simulation data. Direct comparison of simulated and measured data leads to verification of the device and the test environment.

Simulations are valuable tools to understand theory by implementing equations and exploring variations. For circuit simulation SPICE variants (LTSPICE, PSPICE, Multisim) are used. Due to the large number of bits low level circuit simulation of data converters is very slow and requires a lot of memory and computing power. High level simulations are used to estimate data converter performance and understand conversion limitations. For high level simulation tools like MatLab Simulink or a programming language like C are used. These tools are expansive and it is difficult to document simulation setup and results. This paper presents a low cost serial charge sharing DAC: schematic, high level simulation, practical realization, measurement and characterization.

A serial DAC is a simple, low component count easy to understand DAC [4]. Figure 1 shows a serial DAC (C1, C2, CLK1, CLK2) with sample and hold (X1, X5, C3, CLK3).

Data is serially latched with CLK1 in C1 (LSB first). Charge is shared between C1 and C2 with CLK2 activated. This cycle is repeated for each bit (serial). At the beginning CLK1 and CLK2 are active to put 0 V on both capacitors. The sample and hold circuit (preamplifier operational amplifier X1, switch CLK3 X5, operational amplifier X6) stores the output voltage. This circuit in principle could have infinite resolution for infinite number of charge sharing cycles. To calculate the final output voltage a charge computation is done [cf. equations (1) – (6)].

$$Q = C \cdot V \tag{1}$$

$$C2 \cdot Vx2(n) + C1 \cdot D(n) = Vx2(n+1) \cdot (C1 + C2) \tag{2}$$

$$Vx2(n+1) = (Vx2(n) \cdot C2 + C1 \cdot D(n)) / (C1 + C2) \tag{3}$$

with $C1 = C2$

$$Vx2(n+1) = 0.5 \cdot Vx2(n) + 0.5 \cdot D(n) \tag{4}$$

$$Vx2(n+1) = 0.25 \cdot Vx2(n-1) + 0.25 \cdot D(n-1) + 0.5 \cdot D(n) \tag{5}$$

$$Vout = \sum_{k=0}^{nBit-1} 2^{nBit+k} D(k) \tag{6}$$

The equations show the binary weight of the data inputs. This is a binary weighted DAC having a very low component count. Practical limits of this circuit are the precision of the capacitances C1 and C2, the leakage of the switches (back bias of transistors) and the leakage, offset and non linear gain of the operational amplifier. The precision of the matching of the capacitances have to be below 1 LSB. For a 16-bit DAC this would be $1/65536 = 15\,2$ppm. Thermal noise of the R of the switches and operational amplifiers noise are also limiting the maximum resolution.

In the following sections first the LTSPICE simulation circuit with input signal generation for ramp and sine signals is presented. Open access internet pages are used to read and evaluate LTSPICE data [5]. The code for these pages is freely available at no cost. Due to the slow LTSPICE simulation and to model DAC errors properly an internet based JavaScript high level simulation is presented. All internet pages can also be run locally on a computer and source code can be modified to study other effects and architectures. Next the circuit is realized using an operational amplifier, NFET and PFET arrays and an Arduino board. Realization can prove the match between theory, simulation and real circuit and can highlight missing model assumptions. Then measurements of ramp and sine signals give INL, DNL, FFT, SNR and ENOB. Errors showed additional challenges of a real circuit and lead to error correction to improve ENOB. Finally results are discussed.

II. LTSpice Simulation

For simulation with LTSPICE (no cost program) serial digital data for a ramp and sine signal have to be generated. An ideal model of a 4 bit pipeline ADC and a switch matrix is used to generate the appropriate input signals easily. Figure 2 shows the 8-bit serial DAC circuit. An ideal DAC is omitted on the picture, but implemented to have an ideal output voltage for comparison. The resolution is limited to 8-bit to get reasonable simulation times. Real transistor

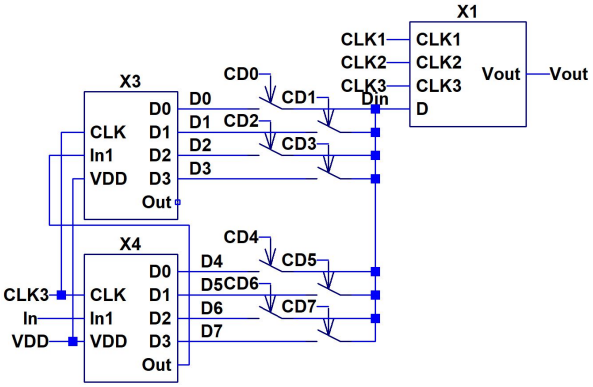

Figure 2. LTSPICE test circuit.

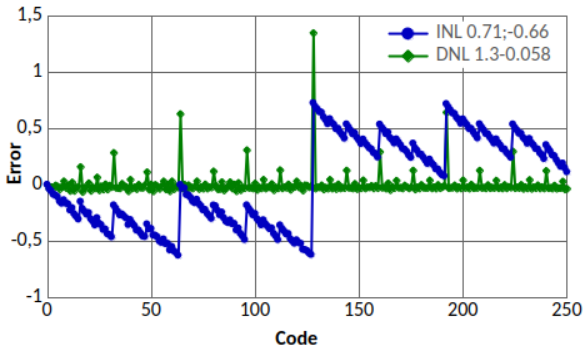

Figure 3. INL and DNL of LTSPICE ramp simulation.

models and operational amplifier models are used. Low frequency clock signals ($t_{CK} = 60\,\mu$s) are used since the practical circuit is also operated at low speed.

Care has to be taken to limit the raw output data file size with a save statement and to control precision with *.option plotwinsize=0*.

First an up/down ramp is used to have minimum settling time for accurate output values. The DAC output comes one cycle later than the input data. Care has to be taken that all codes are exercised and generate an output value. The circuit for download and simulation is available on a website [5].

A. Ramp simulation for INL and DNL

Figure 3 shows the INL, DNL of LTSPICE simulation (10 MB file size) after processing with JavaScript [5] tool in the browser.

LTSPICE writes data with varying step size. Output data has to be filtered with sampling times. From this raw ramp data minimum and maximum is extracted. Real step size LSB is calculated in (7).

$$\text{LSBreal} = (\text{Maximum} - \text{Minimun})/\text{NSteps} \tag{7}$$

An ideal curve compensating for offset (Minimum) and gain error is generated (8):

$$\text{OUTideal(i)} = \text{Minimun} + \text{i} \cdot \text{LSBreal} \tag{8}$$

and INL and DNL are computed ((9) and (10)):

$$\text{INL(i)} = (\text{OUT(i)} - \text{OUTideal(i)})\,/\text{LSBreal} \tag{9}$$

$$\text{DNL(i)} = (\text{OUT(i)} - \text{OUT(i-1)} - \text{LSBreal})\,/\text{LSBreal} \tag{10}$$

All these steps are automatically done by JavaScript at client side in the browser. A graph is generated as displayed in Figure 3. Maximum DNL of 1.5 and INL minimum of -0.8 can be seen reducing the effective number of bits. The program works up to a few million data points.

The circuit has a big error step at MSB switching limiting the resolution of the DAC. The effective number of bits is reduced.

The pattern of INL and DNL is typical for binary weighted data converters [6]. The DNL has a maximum at the center and can be up to 2 times the INL maximum error. DNL peaks can be seen only at multiples of power of 2 bit switching operations. Higher order bits contribute more to DNL error. A mismatch between the capacitances C1 and C2 or added capacitance of the CMOS transistor switches can lead to this pattern.

B. Sine simulation for FFT and SNR

An 8-bit DAC needs at least $256 \cdot 4$ points for FFT due to changing slope of a sine signal and an odd or prime integer number of periods to exercise all codes.

Having an integer number prevents FFT bleeding and windowing can be omitted. The simulation time t_{meas} is calculated as given in (11).

$$t_{\text{meas}} = 1024 \cdot 480\,\mu\text{s} = 491.52\,\text{ms} \tag{11}$$

Having $N_{\text{period}} = 13$ periods gives the signal frequency calculated in (12).

$$f_{\text{signal}} = N_{\text{period}}/t_{\text{meas}} = 26.448\,567\,71\,\text{Hz} \tag{12}$$

The raw output data is evaluated with the same web page as ramp data. Start time $0\,\text{s}$, stop time $491.7\,\text{ms}$ and time step $480\,\mu\text{s}$. Figure 4 shows the resulting FFT mapping the DC content to a frequency of $0.1\,\text{Hz}$. Numbers for signal and noise level are shown at the top of the graph. Total noise level is shown as a line. The number of FFT points can be easily extracted from the maximum frequency as $2 \cdot 512 = 1024$. To be able to asses an FFT the number of points determine the total noise, since all noise bins have to be summed up for total noise.

The signal can be seen at frequency $13\,\text{Hz}$ with $-3\,\text{dB}$ magnitude, total noise is $50.14\,\text{dB}$ and biggest harmonic HD3 at $-55.67\,\text{dB}$ below an expected total noise level calculated in (13).

$$-3\,\text{dB} - 1.76\,\text{dB} - 8 \cdot 6.02\,\text{dB} = -52.92\,\text{dB} \tag{13}$$

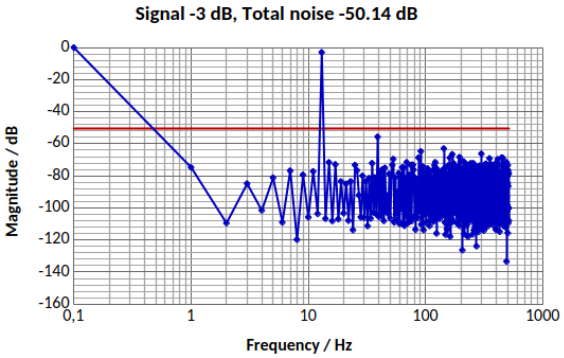

Figure 4. FFT of the simulated serial C DAC.

Table I
FFT MAGNITUDE RESULTS

Frequency Hz	Signal magnitude dB	Total noise magnitude dB
0	$-0,04$	$-3,00$
13	$-3,00$	$-50,14$
39	$-55,67$	$-51,65$
143	$-63,56$	$-51,94$
299	$-65,98$	$-52,11$
91	$-66,28$	$-52,28$

The total noise level can not be extracted easily from the FFT graph and needs some computation which is done by the web page in JavaScript at the client. Without the highest harmonic HD3 the total noise level is $-51.65\,\text{dB}$. The $-3\,\text{dB}$ shows the rms value of the $1\,\text{V}$ amplitude. The tool gives a table with the frequencies in order of signal magnitude (Table I).

This allows to look at the signal to noise level ($47.29\,\text{dB}$) and the impact of harmonics. The effective number of bits (ENOB) can be calculated from Table I as given in (14) and (15).

$$\text{ENOB} = \frac{\text{signal lvl} - (-\text{total noise lvl}) - 1.76\,\text{dB}}{6.02\,\text{dB}} \tag{14}$$

$$\text{ENOB} = \frac{-3\,\text{dB} - (-50.29\,\text{dB}) - 1.76\,\text{dB}}{6.02\,\text{dB}} = 7.56 \tag{15}$$

The ideal sine signal with $3\,\text{V}$ amplitude at the LTSPICE input with the ideal DAC, which was simulated as sanity check for comparison, shows no visible harmonic and gives the effective number of bits according to (16).

$$\text{ENOB} = \frac{0.51\,\text{dB} - (-49.1\,\text{dB}) - 1.76\,\text{dB}}{6.02\,\text{dB}} = 7.78 \tag{16}$$

The first highest harmonic for an ideal signal is at frequency $447\,\text{Hz}$ with $-65.33\,\text{dB}$. The simulation shows harmonics not limiting ENOB but the total

Table II
FFT MAGNITUDE OF BINARY DAC WITH SYSTEMATIC ERROR

Frequency Hz	Signal magnitude dB	Total noise magnitude dB
13	−9.04	−54.88
39	−59.57	−56.69
299	−68.10	−57.01
91	−69.44	−57.27

noise. LTSPICE ideal DAC accuracy is still not showing the full ENOB = 8.

Based on this LTSPICE model circuit improvements and characterization can be easily done at no cost. Unfortunately for larger number of bits the simulation time and data size becomes quite big. A 16 bit simulation stopped after 84 % of simulation after 10 h and 1.6 GB file size due to lack of virtual address space on a system with 16 GB main memory. A high level simulator is needed for faster analysis. A high level simulator shows also understanding of error sources by implementing the equations correctly.

III. JAVASCRIPT DAC SIMULATION

An Internet webpage [5] with a Javascript simulator was created to be able to study DAC errors faster (Figure 5). Unit or binary element DACs with different number of bits can be selected with random or systematic element error. Additionally a random noise measurement error can be added. Typical INL, DNL charts are generated. For FFT the number of points and number of periods of a test signal can be selected. Sine signal can have non integer and non prime number of periods. Effects on FFT with and without windowing can be studied. This can give a test engineer a feel for typical measurement errors and impact on results. Figure 5 shows the user interface and a result for an 8 bit binary element DAC with systematic error of −0.0065. INL and DNL is provided for one random error run and absolute INL and DNL maxima for a user defined number of runs. The QR code allows to access the internet page directly. Also some error correction method is implemented as presented later.

The typical INL, DNL pattern of a binary element DAC can be seen again and the FFT showing a harmonic. The values were chosen to match the LTSPICE simulation. A table with FFT magnitude values is also generated (Table II).

The SNR of the simulation is $-9.04\,\mathrm{dB} - (-54.88\,\mathrm{dB}) = -45.84\,\mathrm{dB}$ a little less than LTSPICE simulation. The first biggest visible harmonic (39 Hz) is still below total noise level. The next figures (Figure 6, Figure 7) show the results for a random error of 0.01 for unit element and binary element DACs. The INL graph shows also the maximum of INL and DNL of 16 different random runs giving a worst case scenario.

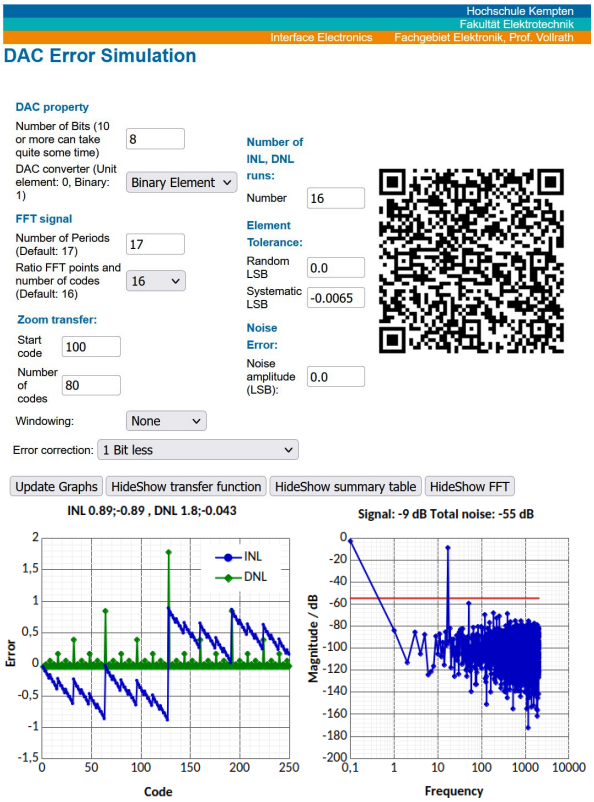

Figure 5. Javascript Internet Page with DAC Error Simulation.

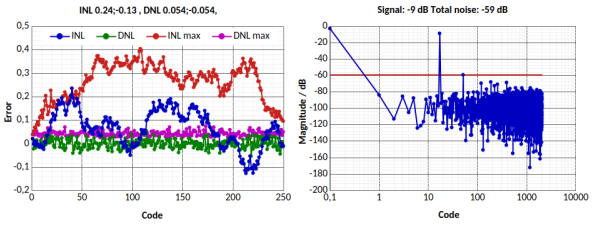

Figure 6. Random unit element DAC Error simulation.

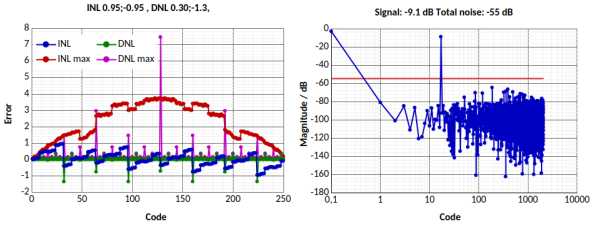

Figure 7. Random binary element DAC Error simulation.

The simulation results change each run time due to the randomness of errors. It is possible to see typical patterns due to the architecture. INL of the unit element DAC can have one or more bumps, DNL is much lower and more random.

DNL peaks are much bigger at the binary element DAC and INL shows a sawtooth pattern. FFT of the binary element DAC show a high harmonic due to the high DNL value. It is very easy to locate errors of the

Table III
SYSTEMATIC ERRORS, INL, DNL, SNR AND SDR

Type	Systematic Error	INL DNL max	Magnitude Harmonic dB	SNR w/o Harmonic dB
Unit	0.00001	0.08	HD2: -79	49.74
Unit	0.0001	0.8	HD2: -59	46.86
Binary	0.002	0.48	HD3: -76.24	49.50
Binary	0.01	2.2	HD3: -59.52	45.31

binary element DAC looking at the DNL and to correct them. Positive DNL determine the minimum step size, negative DNL causes code loss which can be corrected.

Only the binary element FFT shows visible harmonics. The bumps of the unit element DAC are too small to generate harmonics. It is interesting to vary the random error and look at many runs to observe the visibility of harmonics. The total noise will always be affected by INL, DNL error. The total noise of a binary element DAC is bigger than the unit element DAC with the same random error.

A systematic error can be applied to look at typical worst case INL, DNL and FFT patterns. Table III shows a summary.

Unit element DACs with systematic errors generate 2^{nd} hamonics, binary element errors generate 3^{rd} harmonics. Since there are more unit elements needed than binary elements smaller systematic errors can cause a harmonic.

IV. DAC CIRCUIT MEASUREMENT

To allow comparison between theory, simulations and measurement the circuit was built and measured. For cost reasons an Arduino board was used for control and data acquisition. It is very cheap, easy to program, has a 12 bit DAC and ADC for voltage acquisition and reference and enough digital pins to operate the switches and data input at 3.3 V. Measurement data is transfered via the serial interface to the PC. The sampling speed is very slow.

The serial DAC needs only a few components and is very simple. It was build with one opamp IC 272, one ALD1106 and one ALD1107 integrated circuit with 4 NFET and 4 PFET transistors each as switches and 3 10 pF capacitances. Beside the 3.3 V the operational amplifier needs extra voltages (5 V, 1.5 V) which were provided via battery or an Electronic Explorer power supply to amplify a signal in the 0 V to 3.3 V range. Figure 8 shows the final set up.

Data processing is done via another JavaScript tool. It is planned to have a local nodejs server controlling the Arduino via serial interface and providing a better web interface for the user. This would improve characterization capabilities. The Arduino program is also available on the web site.

The internal waveforms for circuit verification shown in Figure 9 were measured with an oscilloscope.

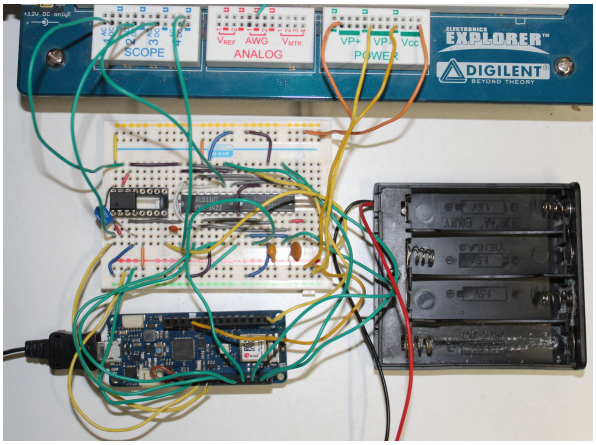

Figure 8. Serial C DAC circuit realisation with Arduino and power supply.

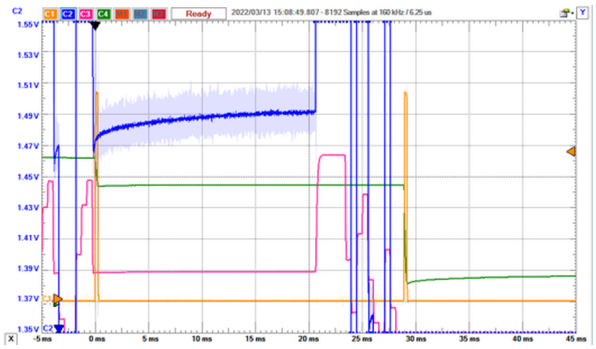

Figure 9. Internal signals of serial DAC.

C1 (yellow) shows the sampling every 29 ms. C3 (red) shows the internal node of the second capacitance C2 (blue). After a long initialization of 20 ms the data is transferred serially to the capacitance. C2 is a zoomed waveform of C3, which shows charging and discharging of this capacitance during the charge equalization times probably due to leakage. This is one reason for the steps. C4 (green) shows output voltage changing at the sampling with CLK3 and C1 signal.

The breadboard allows easy circuit modification and access to all signals. The Arduino environment provides easy change of CLK signal sequence, voltage levels and provides data transfer to a PC.

Since the basic functionality was confirmed, noise measurements were done and compared to averaging results (Table IV). Averaging of 16 values at code 2040 gave an average value of 2238 and shows a good compromise between accuracy and acquisition time. 64 would give a delta below 1, but was not used in this work due to time constraints. An automatic test sequence would allow application of higher averaging values. Theory states a gain of 10 log(OSR) in resolution for averaging. This can be seen in the reduction of standard deviation by a factor of 2 for an oversampling ratio of 4.

Table IV
AVERAGING OF SAMPLES AND DELTA FOR CODE 2040

	Samples				
	1	4	16	64	256
Min	2223.0	2234.0	2237.4	2238.0	2238.3
Max	2250.0	2241.8	2239.7	2238.9	2238.5
Delta	27.00	7.75	2.31	0.91	0.16
Npoints	1024	256	64	15	4
Standard Deviation	2.33	1.21	0.54	0.27	0.07

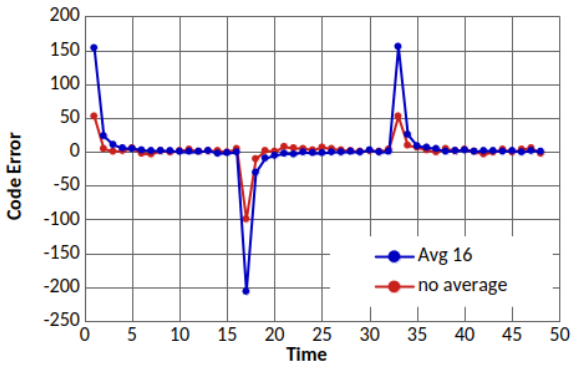

Figure 10. Code error for full scale settling time.

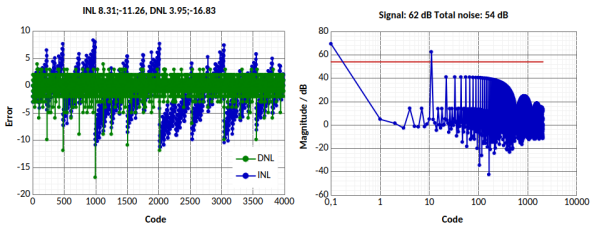

Figure 11. Raw measurement INL, DNL and SNR.

Next full scale and mid scale settling time was measured. For full scale settling time all bits are changing from 0 to 1 or vice versa. Figure 10 shows only the difference to final values of this measurement with and without averaging.

At least 2 samples are needed until the final value is reached. It is expected that large voltage changes will effect measured SNR of this set up. The circuit has some memory effect and should be improved. On the other hand this error can be measured and used in studying error correction.

Next ramp measurements were done and offset (20 codes) and gain errors observed. The output started with a code of 20 and reached 4095 for a input code of 4040. The full resolution of 12 bit could not be reached in this set up with these values since codes 20+55 are missing.

First ramp measurement for 12 bits gave INL between 8.31 and -11.26 and DNL between 3.95 and -16.83. FFT gave 62 dB signal and 54 dB total noise as seen in Figure 11.

Table V
FFT MAGNITUDE OF BINARY DAC WITH SYSTEMATIC ERROR

Min	Max	Signal Magnitude dB	Total Noise Magnitude dB	Cutoff
100	3899	62.56	-10.60	0
400	3500	61.72	40.17	300
200	3799	62.44	28.64	100
132	3868	62.54	17.13	32
116	3883	62.55	9.94	16
108	3891	62.56	2.79	8
104	3891	62.56	-3.77	4

Range matching is one important feature of DAC testing. Therefore it is interesting to see what happens if the sine signal is overshooting and top and bottom are cut off. Table V shows a typical evaluation. A cut off of 4 codes affects the signal to noise behaviour by 6 dB. During test only a slightly smaller range than minimum and maximum code should be tested.

V. ERROR CORRECTION

A calibration or error correction can be done based on the ramp INL, DNL data. Here 2 schemes are discussed. First only 31 code changes of multiples of 128 for the 5 MSB bits are corrected. Secondly a lookup table is constructed. Error correction always reduces the number of possible codes. Therefore the sine range and offset have to be adjusted.

Due to the pattern of the binary DAC transition voltages can be defined where an offset has to be added to the DAC code value. In this example of a 12-bit DAC 7 correction values were added for the 3 MSBs.

The correction values $C(i)$ can be extracted from the ramp measurement and added $C_{\text{sum}}(i)$. A comparator compares the input code with the transition voltages $V_t(i)$ and applies the correction values $C(i)$. For $V_t(i)$ the sum of the correction code has to be subtracted. In this example the 3 MSB give the transition voltages:

$$V_t = 511, 1023, 1535, 2047, 2559, 3071, 3582$$

From the ramp measurement the following correction factors were taken:

$$C = 6, 10, 5, 12, 4, 8, 5$$

At each transition voltage the correction factors are added up:

$$C_{\text{sum}} = 6, 16, 21, 33, 37, 45, 50$$

Each transition voltage changes due to the $C_{\text{sum}}(i)$:

$$V_t = 511, 1017, 1519, 2014, 2522, 3026, 3532$$

In the control program the input code is compared with the transition voltages and correction $C_{\text{sum}}(i)$ is added. 3903 Codes remain, DNL and INL are improved (Figure 12).

Secondly lookup calibration can be done. A lookup table is created using sorted pivot operation on ramp

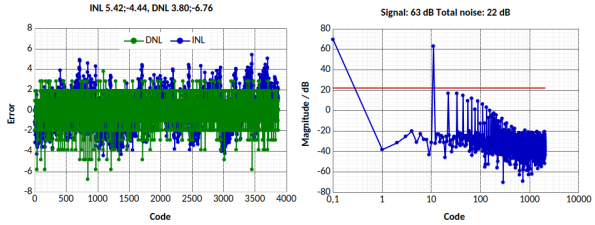

Figure 12. Improved custom calibration INL, DNL, and SNR.

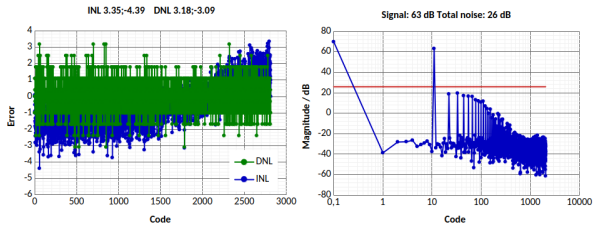

Figure 13. Improved lookup calibration INL, DNL and SNR.

data displaying for each output code the maximum input code. Then a lookup for each possible input code is done for the maximum input code. In this example 2840 unique codes remained (Figure 13).

Table VI shows the results for ENOB INL and DNL for an ideal 12-bit sine signal, the uncalibrated serial DAC sine signal, custom and lookup calibrated DAC sine signal.

It was also investigated how the number of FFT points NFFT is affecting signal to noise ratio (Table VII).

More FFT points decreased the frequency of the signal increasing the ENOB. This was confirmed measuring with different frequencies (Table VIII).

As expected from the full settling time measurement big changes in voltage present in high signal frequency measurements can cause this. This would also be a typical behaviour in the presence of jitter. Since the board is generating the signal synchronized to the the ADC data acquisition there should be no jitter present. More research in this area to improve the circuit is needed.

VI. Summary and Outlook

A binary element serial C DAC is presented with theory, low level, high level simulation, circuit realization, analysis tools and characterization. Typical characterization steps were done using online web based JavaScript internet tools for analysis. The circuit is very cheap to built and shows enough errors to be an interesting subject for study.

The acquisition of 1 k data points took 108 s, 60 ms and 30 s for LTSPICE simulation, JavaScript DAC simulation and measurement. With the number of bits the simulation times increase exponentially. A high level simulation is very versatile and fast, but can only model what is found via LTSPICE simulation or measurement. Correlating data confirm understanding of the DAC circuit.

JavaScript tools were also used to read and analyze LTSPICE data. INL, DNL from ramp data and FFT and SNR of sine data are automatically calculated and displayed.

Open access internet based tools make it very easy to analyze data and do simulation. The underlying JavaScript code can be locally copied and modified for different applications and failure modes. Even big data amounts can nowadays be analyzed in the browser in a short amount of time.

All characterization steps with pitfalls were presented: Noise (base code), averaging (increased number of bits), settling time (full scale and mid scale), ramp test (start, stop codes), sine test (offset, range, NFFT, odd or prime number of periods), FFT, SNR and error correction.

Two error correction procedures for binary element DACs were presented in detail and results discussed. The error correction showed a significant improvement in INL, DNL and SNDR. Since the correction procedures were also implemented in JavaScript the code can be studied in detail and improved. The simple circuit shows typical DAC errors and can be easily modified for experiments. This makes characterization of DACs easier and accessable.

Final measurement data showed INL, DNL below 6, SNR of 46.5 dB and ENOB with error correction.

Table VI
Calibration SNR, INL, and DNL

	Ideal Sine	No Calibration	Custom Calibration	Lookup Calibration
Periods	11	11	11	11
Signal Magnitude dB	63.01	62.39	62.92	63.01
Total Noise Magnitude dB	-10.90	54.41	22.27	25.74
ENOB	12.0	1.0	6.5	5.9
INL min	0	-12	-6	-6
INL max	0	8	5	3
DNL min	0	-16	-6	-3
DNL max	0	5	5	6
NFFT	4 k	4 k	4 k	4 k

Table VII
Frequency, Number of FFT Points and SNR

	Ideal Sine	Custom Calibration	Custom Calibration	Custom Calibration
Periods	11	11	11	11
Signal Magnitude dB	63.01	62.88	62.92	62.92
Total Noise Magnitude dB	-10.90	33.03	22.27	11.37
ENOB	12.0	4.7	6.5	8.3
NFFT	4 k	4 k	4 k	4 k

Table VIII
FREQUENCY, PERIODS, AND SNR

Periods	Signal Magnitude dB	Total Noise Magnitude dB	ENOB	NFFT
11	62.00	32.81	4.6	1 k
43	61.68	41.30	3.1	1 k
44	61.68	41.37	3.1	1 k
179	59.42	44.51	2.2	1 k

Improvement of error correction was 32 dB or 5.5 ENOB.

Future work will improve the tool chain and enable more interactive web based DAC characterization. Variations of the circuit to operate from a battery supply and have higher voltage CLK levels are planned to study circuit limitations. More analysis of the circuit is needed to improve the circuit, decrease the error, increase the resolution and speed up the sampling frequency.

REFERENCES

[1] M.P. Tiilikainen. "A 14-bit 1.8-V 20-mW 1-mm/sup 2/ CMOS DAC". In: *Solid-State Circuits, IEEE Journal of* 36 (Aug. 2001), pp. 1144–1147. DOI: 10.1109/4.933474.

[2] Jen-Huan Tsai et al. "A 14-bit 200MS/s current-steering DAC achieving over 82dB SFDR with digitally-assisted calibration and dynamic matching techniques". In: *Proceedings of Technical Program of 2012 VLSI Design, Automation and Test*. 2012, pp. 1–4. DOI: 10.1109/VLSI-DAT.2012.6212594.

[3] D.A. Mercer. "Low power approaches to high speed CMOS current steering DACs". In: Oct. 2006, pp. 153–160. DOI: 10.1109/CICC.2006.320868.

[4] Jesper Steensgaard, U.-K Moon, and Gabor Temes. "Mismatch-shaping serial digital-to-analog converter". In: vol. 2. Aug. 1999, 5–8 vol.2. ISBN: 0-7803-5471-0. DOI: 10.1109/ISCAS.1999.780472.

[5] J. Vollrath. *Low cost serial DAC simulation, realization, error correction and characterization*. URL: https://personalpages.hs-kempten.de/~vollratj/Projekte/CSerial.html (visited on 10/07/2024).

[6] Chi-Hung Lin and K. Bult. "A 10-b, 500-MSample/s CMOS DAC in 0.6 mm/sup 2/". In: *Solid-State Circuits, IEEE Journal of* 33 (Jan. 1999), pp. 1948–1958. DOI: 10.1109/4.735535.

Joerg E. Vollrath received 1989 his Dipl. Ing. and 1994 his Ph. D. in electrical engineering, semiconductor technology at the University of Darmstadt, Germany. Since then he worked for the memory division of Siemens and Infineon Technologies and Qimonda in various locations in the USA and Germany. He is now a Professor for Electronics at the University of Applied Sciences, Kempten, Germany. His expertise and interests lie in the field of design of analog and digital circuits, programmable logic, test, characterization, yield, manufacturing and reliability. He has published 22 papers and holds 23 patents.