

# A parallel current-steering DAC architecture for flexible and improved performance

R.A.T. van den Hoven, G.I. Radulov, J.A. Hegt, A.H.M. van Roermund.

Mixed-signal Microelectronics Group, Eindhoven University of Technology, Electrical Engineering, Eindhoven, The Netherlands, Email: HovenR@gmail.com

**Abstract**— This paper presents a new current-steering DAC architecture for flexible and improved performance. This flexible DAC architecture is based on fixed entities: sub-DACs. They are nominally identical and operate in parallel, which results in improved and flexible performance, delivered in several modes of operation (OP). One OP mode is using the sub-DACs as independent converters. Another option is using them together for higher conversion resolution and accuracy. This paper concentrates on a particular OP mode, which through distributing the input digital word among the parallel sub-DACs, achieves cancellation of the mismatch errors. This technique leads to improved static linearity, whereas the improvement depends on the occupied pre-processing resources. The proposed technique can be fully integrated on-chip, as it relies on a 1bit ADC and makes reuse of already existing resources.

**Index Terms**—Digital-to-Analog converters, Flexibility

## I. INTRODUCTION

Current-steering DACs are systems composed of three main parts. These are the digital control part, the analog unit elements, and the switches in the mixed signal part, see Figure 1. In conventional approaches, all these three parts are fixed. Thus, the performance is fixed and unique. Once a conventional DAC is manufactured, it has a certain resolution and accuracy. Its non-linearity is unique, because the mismatch errors of its current sources and their distribution are unique.

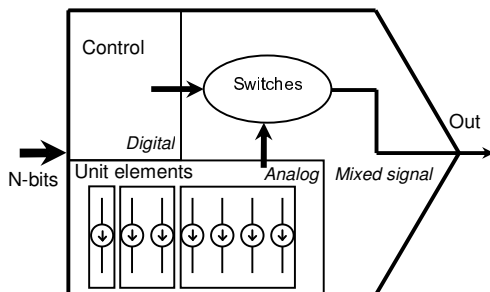


Fig. 1. DAC with grouped unit elements.

Popular approaches to improve this unique static performance include:

- intrinsically accurate current sources;
- calibration;
- digital pre-processing.

The work is sponsored by Stichting Technische Wetenschappen.

The first two approaches reduce the mismatch errors, while the third approach alters the distribution of the errors. Important advantages of digital pre-processing include that:

- the D/A process is intact;
- the needed extra resources are in the digital domain.

Our approach uses an architecture based on parallel sub-DACs, shown in Figure 2, [1-3]. Extra digital pre-processing controls the fixed sub-DACs. The sub-DACs can be used in different OP modes and hence the resolution and accuracy are no longer fixed but flexible. Note that the flexibility in the digital part affects the whole DAC without a need for major changes in the analog and mixed-signal parts.

In this paper, a technique is presented to improve the linearity of the parallel DAC system. In section II the definition of a parallel DAC architecture and the principles for linearity improvement are presented. Section III presents a self-measurement and self-correction method. The algorithms and required memory are explained in section IV and V. Finally, simulation results and conclusions are given in Section VI and VII.

## II. PARALLEL DAC ARCHITECTURE

Some of the OP modes combine the sub-DAC outputs to construct one single higher resolution DAC, as shown in Figure 2.

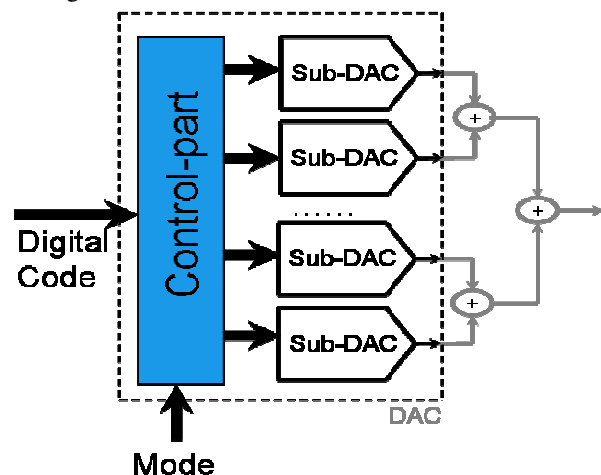


Fig. 2. Parallel DAC architecture.

When the end-user requires higher accuracy, our flexible DAC can distribute the digital input code among its sub-DACs, in such a way that their individual mismatch errors cancel each other. Note that the outputs of the sub-DACs have to be combined at the outputs of the chip.

With a single DAC there is only one output combination for each input code. When two DACs are used the number of output combinations increases. For example, with two sub-DACs there are three combinations to construct digital code 2, etc.. Table 1 shows the possible combinations to construct the analog output for all digital codes in the case of two parallel 11bit sub-DACs.

Input	Combinations to distribute codes	Ck
0	0+0	1
1	0+1, 1+0	2
2	0+2, 1+1, 2+0	3
3	0+3, 1+2, 2+1, 3+0	4
-		
2046	0+2046, 1+2045, ---, 2045+1, 2046+0	2047
2047	0+2047, 1+2047, -----, 2047+1, 2047+0	2048
2048	1+2047, 2+2047, ---, 2047+2, 2047+1	2047
-		
4092	2045+2047, 2046+2046, 2047+2045	3
4093	2046+2047, 2047+2046	2
4094	2047+2047	1
4095	-	0

Table 1. Combinations with two 11-bit sub-DACs.

Note that for every extra sub-DAC one input code is missing. This is because every sub-DAC has  $2^N$  codes and only  $2^N-1$  current sources. The number of input codes for the combined DAC with M N-bit sub-DACs is:

$$Codes_{in} = M \cdot 2^N - (M - 1) \quad (1)$$

The number of sub-DACs affects the number of combinations for each code, shown in Figure 3.

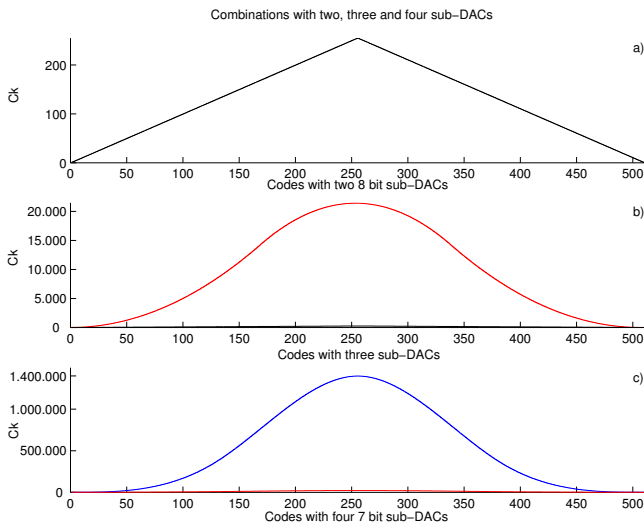


Fig. 3. Number of combinations with two, three and four sub-DACs.

The number of combinations with two sub-DACs (shown in Table 1 and Figure 3a) for digital code  $k$ , with  $k$  from zero to a half-scale is:

$$C_{k2} = k + 1 \quad (2)$$

The number of combinations with  $k$  from a half- to full-scale is:

$$C_{k2} = (fullscale - k) + 1 \quad (3)$$

The total number of possible combinations with “M” multiple N-bit sub-DACs is:

$$C_M = \sum_{k=0}^{M \cdot N - (M - 1)} C_k = (2^N)^M = 2^{M \cdot N} \quad (4)$$

To construct the combined DAC transfer characteristic two sub-DACs have  $2^{2N}$  combinations to choose from for  $2^{N+1}-1$  codes (according to equations 4 and 1). When for each code the combination with the best linearity is selected, the  $INL_k$  (the INL error for code  $k$ ) is equal or smaller than with a random/fixed combination. This is shown for four sub-DACs in Figure 4b.

With each additional sub-DAC the number of combinations, and hence the chance to find a better one increases, as shown in Figure 3b and 3c. The total number of combinations for 3 and 4 N-bit sub-DACs are  $2^{3N}$  and  $2^{4N}$  (Equation 4) and there are only  $3 \cdot 2^N - 2$  and  $4 \cdot 2^N - 3$  codes to construct (Equation 1). Note that the INL improvement will be less at the ends of the transfer characteristic, because there the number of possible combinations is smaller. At half-scale there are the most possible combinations and thus the possibility to create one for a better linearity is higher.

The difference in linearity with a fixed and flexible architecture is shown in Figure 4. For each simulation an 8-bit DAC is constructed with four 6-bit sub-DACs. In Figure 4a, the control of the sub-DACs is fixed, i.e. the distribution of the digital codes is not optimized for improved linearity.

Like in the normal DACs, the highest statistical possibility of locating  $INL_{max}$  is at mid-scale. Figure 4b shows the INL of the same 8-bit DAC but with full flexibility in the control-part and linearity optimization through choosing the best code combination to generate as small INL error as possible. With four sub-DACs a combination can be found that creates small  $INL_k$  errors. However, when there are many combinations, there is always a very small probability that none of the combination creates a very small  $INL_k$ . With the 700 (Monte Carlo) simulations shown in Figure 4b this happened only once around code 220.

To improve the linearity of the D/A transfer characteristic the transfer characteristics of the sub-DACs are acquired. The best

combinations of “sub-  $INL_k$ ’s” can reduce the nonlinear effect caused by the mismatches. This will improve the overall INL, as shown in Figure 4b.

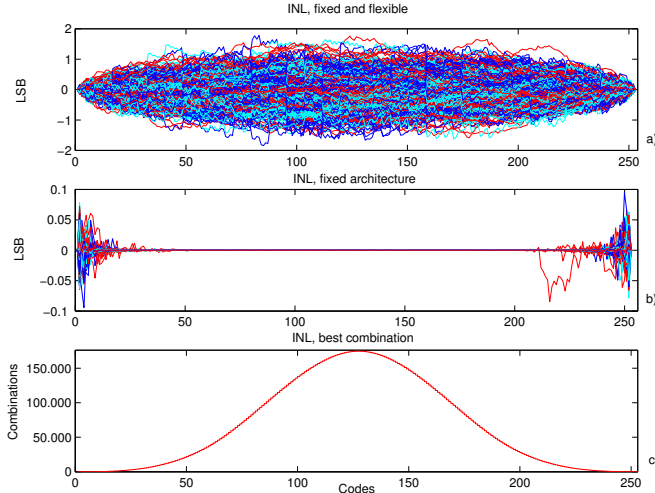


Fig.4. INL improvement with four sub-DACs in a fixed and flexible architecture.

- a) INLs of fixed DACs,  
b) INLs of flexible DACs with the same unit elements as the fixed DACs,  
c) Number of possible combinations for each code.  
Monte Carlo simulation.

### III. PARALLEL SELF-MEASUREMENT AND SELF-CORRECTION

Our parallel architecture has four sub-DACs, one extra current source, a 1-bit ADC and some extra switches. The  $INL_k$ ’s of the sub-DACs have to be determined and stored before a good combination of the  $INL_k$ ’s can be made. The  $INL_k$ ’s can not be directly measured because there is no ideal transfer characteristic to compare with. That is why the  $DNL_k$ ’s are measured and then  $INL_k$ ’s are determined as:

$$INL_k = \sum_{j=1}^k DNL_j \quad (5)$$

#### SELF-MEASUREMENT

To determine the INL of sub-DAC A its  $DNL_k$ ’s are measured. DAC A and B will start with the same code and the difference between the outputs will be “measured” with the DAC C, shown in Figure 5.

The output of DAC-C is downscaled; this means that the output at each code is smaller than the output of DAC A or B. With the 1-bit-ADC it can be detected when the output currents from DAC B+C are sufficiently equal to DAC A. The value of DAC C will be stored in a temporary memory. Next, the code at DAC A will be increased by one, code B stays the same and the extra 1-LSB current source will be switched on, see Figure 5. The difference between the outputs of sub-DAC A and B will be determined again. This difference and the value in the temporary memory are subtracted. Based on this the  $INL_k$  can be calculated, see (5). This will be done for each

code and sub-DAC. The  $INL_k$ ’s are stored in a memory and with this an algorithm (in Section IV) will calculate the best combination of the sub-DACs for each output code.

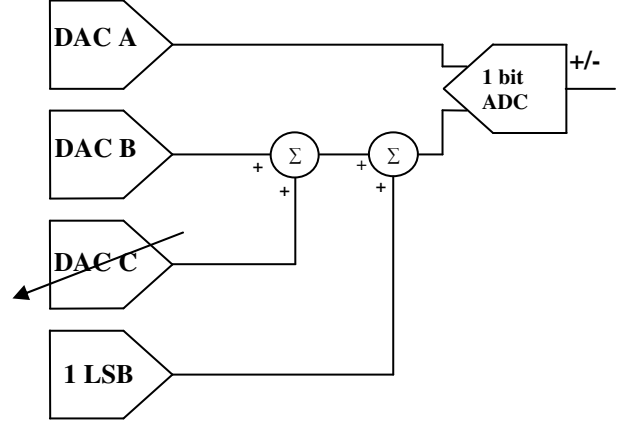


Fig. 5. DNL Measurement diagram.  
Sub-DAC A: To be Measured DAC,  
Sub-DAC B: Comparison DAC,  
Sub-DAC C: Scaled Measure DAC,  
1-LSB: One extra current source.  
1-bit ADC: Current comparator

This self-measurement method uses an extra 1-LSB current-source and this one is not always equal to the average of the DAC current sources. This will introduce an error that is accumulated and affects the stored sub- $INL_k$ ’s. This error can be determined because the  $INL_k$ ’s at full-scale of each sub-DAC should together be zero. If not zero, there is an offset. This offset will affect all the measured sub-  $INL_k$ ’s and the stored  $INL_k$ ’s will be adjusted with the calculated offset.

The measurement-DAC C determines the difference between the output of DAC A and B. Therefore the steps of DAC C have to be smaller than this difference. However DAC C also has to be able to determine the largest possible difference. For example, when the mismatches in each sub-DAC can cause an INL of +/- 2 LSB, DAC C has to be able to measure a maximal difference of +/- 4 LSB. Since in this example single-ended-DACs are used, the negative measurements are done by adding DAC C to DAC A, instead of to DAC B. The INL and quantization errors of DAC C are also accumulated with this method (Equation 5). When the output of DAC C becomes smaller, the accumulation also becomes smaller.

In the simulation, shown in Figure 8, DAC C is constructed with two smaller downsampled-DACs and is able to measure the difference of +/- 4 LSB and has small steps, shown in Figure 6. The first “rough” measure-DAC has  $2^N$  steps and the maximal output is downsampled to 8 normal LSB’s. The second “fine”-DAC can be maximally downsampled to  $2 \text{ LSB}_{\text{rough}}$  of the rough-DAC:

$$LSB_{\text{rough}} = \frac{8}{2^N} = 2^{-(N-3)} LSB_{\text{normal}} \quad (6)$$

$$LSB_{\text{fine}} = \frac{LSB_{\text{rough}}}{2^{N-1}} = 2^{-(2N-4)} LSB_{\text{normal}}$$

The output current in the simulation of the rough-measure-DAC is added to the output current of DAC B and approaches the output of DAC A. The ADC detects when this current becomes larger or smaller than the output of DAC A. The remaining difference will be determined by the fine-measure-DAC. In the simulation, shown in Figure 8, 10-bit sub-DACs are used to construct one 12-bit DAC. The rough-measure-DAC is compared to the normal DAC  $2^7$  times downscaled and the fine-measure-DAC  $2^{14}$  times

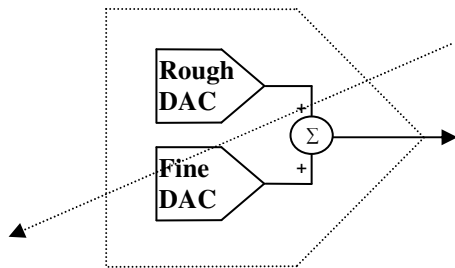


Fig. 6. Measure DAC-C constructed with a rough and fine sub-DAC.

The sub-DACs in a flexible parallel architecture can be used as multiple separate DACs or all/some combined together. When M sub-DACs work together the maximal –combined– output must be the same as the maximal output of a separately working sub-DAC. The individual output currents of the M combined sub-DACs must be M times smaller than normal. When the scalability of the sub-DACs is flexible, the sub-DACs can be used as the rough and fine-measurement-DACs as well. The extra acquired components for this method are the 1-bit ADC, 1-LSB current source, some switches and of course the memory and digital resources for the algorithms.

#### IV. ALGORITHMS TO FIND THE BEST COMBINATION

The best transfer characteristic of the combined DAC can be constructed by using the  $INL_k$ 's of the sub-DACs to find the minimal INL. The best overall INL will be constructed when all the minimal  $INL_k$ 's of the overall code are found. To find the minimal overall  $INL_k$  ( $INL_{kT}$ ) all combinations of the sub  $INL_k$ 's have to be evaluated and every time it finds a  $INL_{kT}$  that is smaller than the  $INL_{kT}$  stored in the memory (at code  $kT$ ), the stored  $INL_{kT}$  will be replaced with the new  $INL_{kT}$ . Also for each input code  $kT$  the sub-codes of the sub-DACs will be stored in the memory as a lookup table. The total number of sub-DAC combinations to evaluate with four sub-DACs is  $2^{4N}$ , according to (4).

For example with four 10-bit sub-DACs there are  $2^{40}$  combinations. With a process that can even evaluate 100.000 combinations per second; the initialization of this DAC would still take 4 months.

A faster algorithm has to evaluate fewer combinations. If some combinations are not evaluated and a few of these combinations would be the best one, these will not be found. However it is expected that with “well chosen” combinations a close to optimal  $INL_{kT}$  will be found. By only actively using two of the four sub-DACs,  $2^{2N}$  combinations can be evaluated (Equation 4). With four sub-DACs there are six options to

couple them. These six options are the combinations of sub-DACs “1&2, 1&3, 1&4, 2&3, 2&4 and 3&4”. When all combinations of those two-coupled-DACs are evaluated, it is expected to find combinations with a small  $INL_{kT}$ . It is clear that with only two active sub-DACs not all of the four sub-DAC current-sources are used. The combined DAC can only use half of the entire code range. To solve this problem the outputs of the passive sub-DACs can be switched to zero or maximal and the active sub-DACs will evaluate the combinations again to find the small  $INL_k$ 's. At the middle of the scale either the first or the second passive sub-DAC can be switched on. This will cause that the number of options to combine the sub-DACs with becomes twice as large at the middle of the scale. The options to combine the sub-DACs are there twelve instead of six times. The middle of the scale options are only used at  $1/4$  of the total code range. The average coupling options for the total code range is:

$$Options_{coupling} = 6 \cdot \frac{3}{4} + 12 \cdot \frac{1}{4} = (6 \cdot 1\frac{1}{4}) = 7.5 \quad (7)$$

The simulated results with this algorithm are still good (shown in Figure 8b) and the total number of combinations is:

$$C = Options_{coupling} \cdot Comb_{DACcouple} = 7.5 \cdot 2^{2N} \quad (8)$$

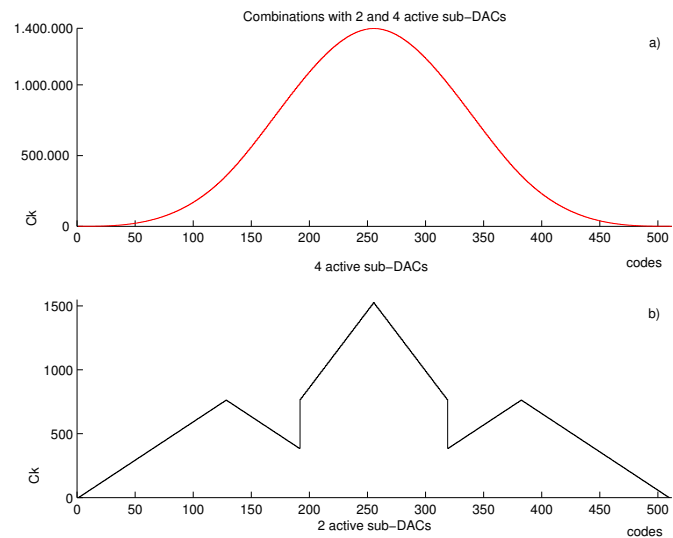


Fig. 7. Number of combinations with four 7bit sub-DACs

a) with 4 active sub-DACs for each code

b) with 2 active and 2 passive sub-DACs for each code.

The simulation, clearly shows peaks at the position where no good  $INL_k$  can be found (Figure 8b). These are at beginning, the end and at the middle. The missing combinations are the main reason of the extra peaks at the middle. It is expected that using an algorithm with more combinations can solve the problems with the peaks at the middle. It is possible to reduce some of the peaks at the beginning and end by using all combinations of four DACs.

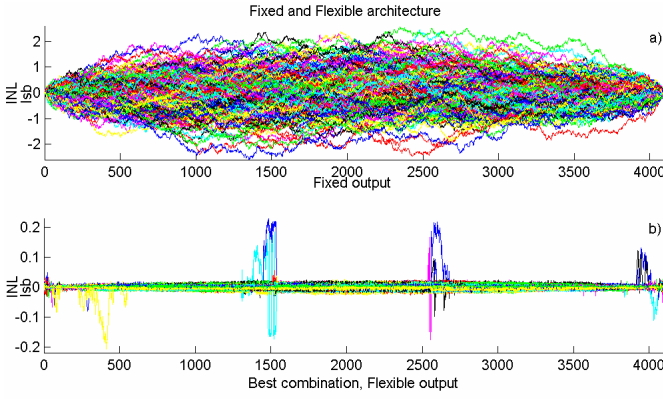


Fig. 8. INL with fixed and flexible architecture. Monte Carlo simulation.

It is expected that the errors caused by the measurement-DACs will have more impact than the missing combinations in the algorithm. The peaks in the middle are caused by 8 of the 200 simulations. It is expected that on average the missing combinations can decrease the performance by 0.5%.

## V. PRE-PROCESSING SELF-CORRECTION

The algorithm of Section IV evaluates the combinations to construct a transfer characteristic and stores the best sub-DAC combinations in the memory as a lookup-table. For each digital input-code the control-part (shown in Figure 2) gets the corresponding sub-codes from the lookup-table and sends these to the sub-DACs.

With four 10-bit sub-DACs, there are  $(4 \cdot 2^{10} - 3)$  4093 input codes (Equation 1). For each input-code the best sub-code combination of the sub-DACs is stored in a memory, this is called the pre-correction lookup table. The memory for the lookup table with four active sub-DACs is:

$$M_4 = \text{Inputcodes} \cdot M_{\text{subDACs}} \cdot N_{\text{subDACs}} \quad (9)$$

$$M_4 = 4093 \cdot 4 \cdot 10 = 163.7 \text{ kbits}$$

The size of the memory can already be decreased by 25% by calculating the fourth sub-DAC code from the input-code and subtracting the other sub-DAC codes. The algorithm of Section IV has only two active DACs and the sub-code of the two passive DACs are zero or maximal, depending on the input-code. The sub-codes of the second active sub-DAC can be calculated from the first one and the input-code, since the third and fourth sub-code are also calculated from it. The lookup table has to remember what are the active and passive DACs are and the sub-codes of the first active DAC. For all input codes there are maximal 12 and on average 7.5 options to combine the active and passive sub-DACs, (according to equation 7). The memory of the lookup table with two active sub-DACs is:

$$M_2 = \text{Input}_{\text{codes}} \cdot (nr_{\text{ActiveDACs}} - 1) \cdot (N + \log^2 \text{options}) \quad (10)$$

$$M_2 = 4093 \cdot (2 - 1) \cdot (10 + 3) \approx 54 \text{ kbit}$$

With this method the memory is decreased by two-thirds. There is also a temporary memory to store the  $\text{INL}_k$ 's of the four sub-DACs (See Section IV),

$$M_{\text{INLsub}} = \text{subcodes} \cdot nr_{\text{DACs}} \cdot N_{\text{subDACs}} \quad (11)$$

$$M_{\text{INLsub}} = 2^{10} \cdot 4 \cdot 10 = 40.96 \text{ kbit}$$

The calculated  $\text{INL}_k$ 's of the combined DAC should also be stored temporarily. However with the two active sub-DACs method it is possible to evaluate all the combinations for each code  $k$  before going to the next code. The same 14-bit memory can be reused and this size is negligible.

The total amount of memory this approach requires is 95kbit. When the lookup table of the DAC is initialized, the 41kbit temporary memory is not needed any more and can be reused.

The size of memory will only become smaller in the future. The new generations FPGAs already have up to 50Mbit internal memory. When parallel DACs are added to these FPGAs, the internal memory and digital resources can be used for the algorithms.

## VI. SIMULATION RESULTS

The simulation with the technique that has two active and static sub-DACs has an average improvement of 97.5% (shown in Figure 8) with a thermometer sub-DAC architecture and 96.5% with a binary sub-DAC architecture. The 1% difference is caused by the larger INL accumulation of the measure DAC with the measurement of the binary sub-DACs. Even with smaller average improvement of the binary architecture, the linear accuracy improvement is:

$$L_{\text{improvement}} = \log_2 \left( \frac{100}{100 - 96.5} \right) = 4.8 \text{ bit} \quad (12)$$

When unit elements with a certain mismatch are used in a fixed architecture, this would have a 10-bit linear accuracy. The unit elements with the same mismatch used in a flexible architecture, a linear accuracy of more than 14-bit can be achieved.

The technique that uses only two active sub-DACs has reduced the total number of combinations more than  $2^{17}$  times.

## VII. CONCLUSION

This paper showed how DAC static linearity can be improved through explicit reordering of the mismatch errors. Parallel sub-DACs are used and the digital input word is optimally distributed among them, so that the mismatch errors cancel each other as much as possible. Simple additional resources and algorithms were used so that a fully integrated solution is possible.

An example of the main concept was shown. Four parallel 10-bit sub-DACs were combined to achieve 12-bit resolution. The initial accuracy was at 10-bit level. After optimizing the distribution of the mismatch errors, the accuracy was improved to higher than 14-bit level.

Our approach is especially attractive in combination with FPGAs. The already existing internal digital resources can be used for pre-processing and only a few extra resources are required. Most of the resources are reused. Our approach will allow DACs with physically smaller current sources, i.e. lower accuracy, because the mismatch errors can be optimally combined and will cancel each other as much as possible.

#### REFERENCES

- [1] P.C.W. van Beek, J.A. Hegt, A.H.M. van Roermund "Optimum segmentation for high speed current steering digital-to-analog converters", ProRISC 2004, Veldhoven, The Netherlands, 2004.
- [2] Eisse Mensink, Eric A.M. Klumperink, Bram Nauta, "Distortion cancellation via polyphase multipath circuits", pp I-1096 – I-1099, ISCAS 2004.
- [3] Deveugele Jurgen, Pieter Palmers, Michiel Steyaert "Single-side-band digital-to-analog converters for Nyquist signal generation", pp 92-96, IEEE 2003.