DCD History

Ivan Peric

CURO

DCD ASICs are based on current mode ADCs. The development of the current mode ADCs for the readout of DEPFET sensors was inspired by the CURO readout chip (designed by Marcel Trimpl in Bonn). The CURO chip used the current memory cells to store the DEPFET currents. In this way the whole signal processing was done in current domain. The CUROs chip did not contain ADCs; it was a binary readout with the possibility to digitize the stored signals (currents) of-chip. The double sampling was possible with the CURO chip – this means DEPFET currents (pedestal+signal) were stored, the DEPFET row was cleared and then the offset currents (pedestal) were stored. In current domain is it easy to subtract two currents. By subtracting of two samples the pure signal current was obtained.

TCUM1

The basic element in the CURO front end was the current memory cell. The cells was quite simple, mainly the cascaded diode connected transistor with the sampling switch before the gate capacitor.

This cells hat tree drawbacks:

1. The transfer characteristics is nonlinear because the Ids = f(Vgs) curve is quadratic
2. The charge injection into the gate when the sampling switch is opened is signal-dependent. This leads to a signal dependent offsets
3. The output resistance of the current memory cell (CMC) is far from optimum (optimum is infinite)

Improved Current Memory Cell

We were, at that time, developing current memory cells which do not have these drawbacks. Our CMC is more complicated – it is based on a differential transconductor (instead of single transistor in CURO) and on an active circuit (amplifier) that keeps the input potential constant, Figure 1.

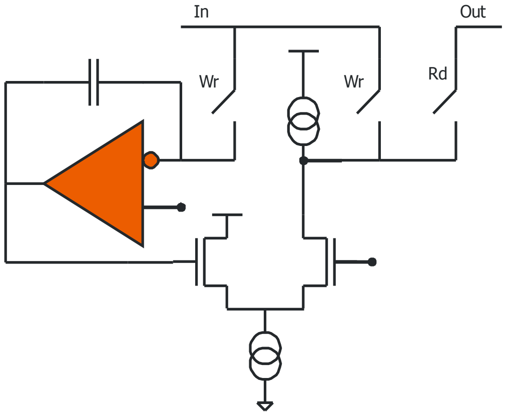

Figure 1: *Current memory cell*

This cell has a nice property that the error due to charge injection is constant – independent of the signal (current) stored in it. (A signal-independent offset is required if we want to do accurate signal processing – such in ADCs.) This also means that the switches in the circuit always see a constant potential at their nodes. If the potential is constant and high enough we can use only PMOS transistors as switches – no NMOS is required. (NMOS are necessary if the potentials at the switch nodes are smaller than about VDD/2, in our case 0.9V.) PMOS-only switches are more radiation tolerant. *NMOS becomes "leaky" after irradiation, which means that we would probably need enclosed NMOS transistors in the switches to make the circuit radiation hard. Enclosed NMOS transistors would inject too much charge into current memory cells leading to large errors.*

Cyclic ADC

Our next idea was to implement the cyclic ADC using such CMCs. The cyclic ADCs are based on the following algorithm: the input signal is compared with two thresholds: one high (ThHi) and one low (ThLo). If the signal is "too high" (>ThHi) a reference signal Ref is subtracted from the input signal. If the signal is "too low" (<ThLo) a reference signal is added to the input signal. The goal of this preprocessing is "to compress" the input signal so that it occupies 2x smaller range. After such signal compression the signal is multiplied by two and a new cycle (signal compression and duplication) is started. In every cycle we obtain two bits of information – the TooHi and TooLow bits. After n cycles we have two binary numbers that are constructed from n TooHi bits and n TooLow bits. We use here the binary representation, the bits generated in the first cycle are MSBs and the bits from the last cycle are LSBs. When we subtract the two

binary numbers we obtain n+1 binary representation of the input signal. In our case n=8 and, after subtraction, we have 9-bit resolution. To simplify the digital transmission we usually discard the LSB of the final result.

Figure 2 shows the transfer characteristics of the signal compression.

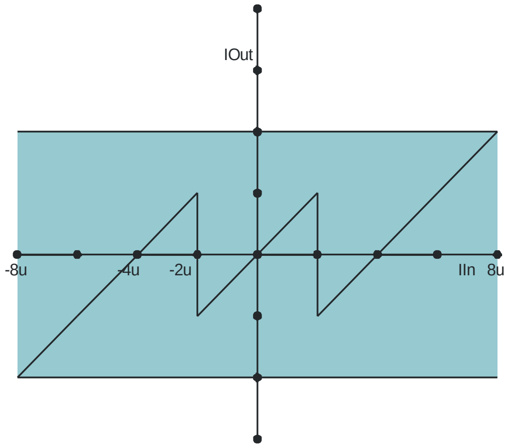

Figure 2: *The transfer characteristics of the signal compression*

There is a very important thing: the result of the AD-conversion depends only on the accuracy of the current duplication and reference subtraction/addition and it does not depend on the threshold values, unless the threshold offsets are higher than 1/8 (12.5%) of the signal range. This can be easily verified by writing down the equations that describe the algorithm – it was also verified in simulations.

In 2005 (?) we have designed a test chip that implements this algorithm using the above described current memory cells. The signal duplication is done by copying of the input signal into two memory cells – i.e. by sampling of the input signal two times. Then the currents can be simply summed together – this summing is equivalent to a multiplication. (Here we assume that the two samples are equal.)

The ADC cycle had three steps. 1) The input current is stored into the first memory cell. At the same time the positive reference current is stored into one auxiliary cell with reversed sign. In this way we obtain the negative reference – if the memorizing of currents is perfect (perfect CMC) the negative reference is exactly the same as the positive one, except of its sign. 2) Input current is stored into second memory cell. At the same time, two copies of the input current from the first cell are made by taking the transconductance amplifier input (voltage) and connecting it to two transconcuctors (TCs) that are identical in layout. The outputs of these two TCs ("copy-TCs") are compared with the thresholds. 3) The currents from cells 1 and 2 and the positive or negative reference current are all added and stored into one result-cell.

From now on a new cycle repeats: The current from the result cell is copied into the first and the second cell and so on.

The algorithm is illustrated in Figure 3.

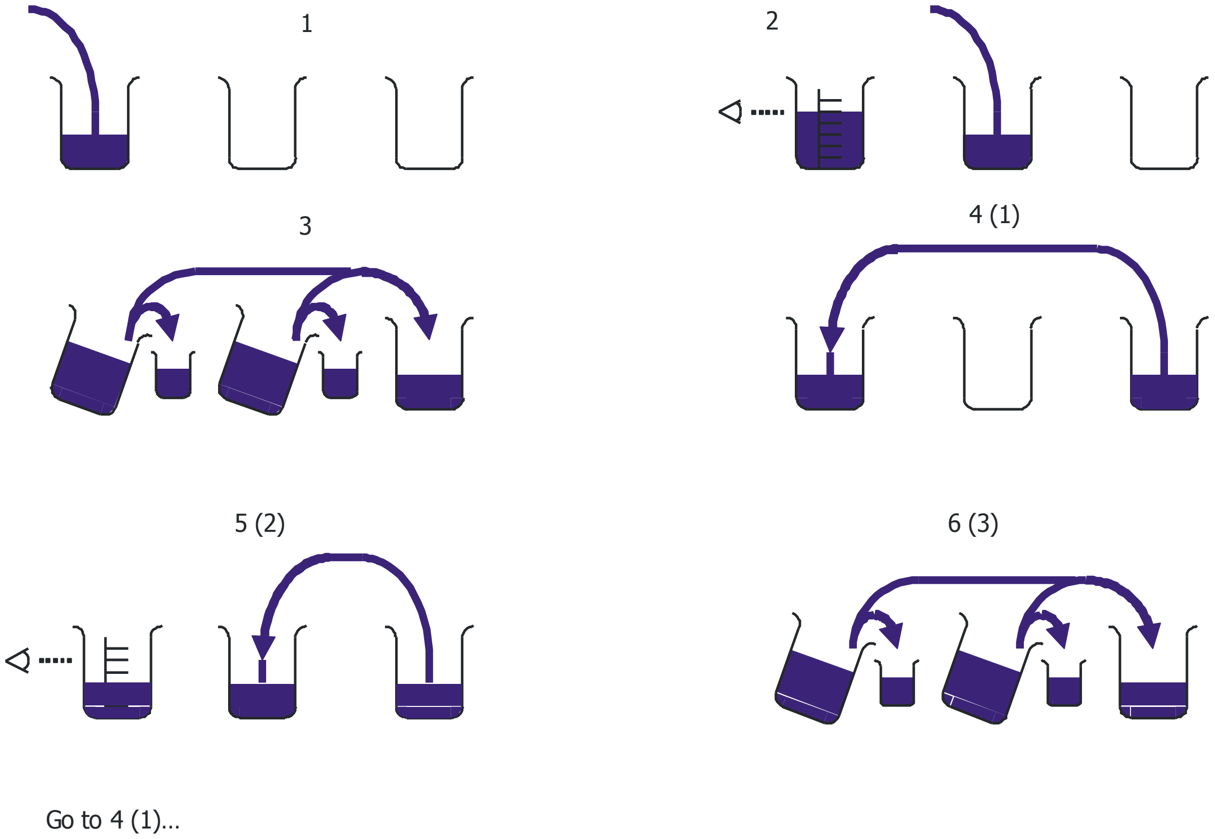

Figure 3: *Three step algorithm*

The ADC worked fine; it was measured within the master work of Tim Armbruster. The drawback of the ADC was that it was slow – it needs three steps per one conversion cycle. The good thing is that the ADC, in principle, relies only on accuracy of the current storing (perfect CMC are needed). It does not rely on any matching between transistors.

However, we did not consider the possible threshold mismatch that can be caused 1) by too fast clocking (comparators do not have time to finish a comparison properly) of 2) by mismatch between TCs (the original and two copy TCs do not match). We did not consider the threshold mismatch because the *requirements are very relaxed – only 12.5% of the full range. This will be an issue at the end of the text.*

TCUM3

In 2006 (?) we have designed an improved version of the ADC (TCUM3) which uses slightly modified algorithm and structure. This structure is used up to now.

The modified algorithm has only two clock periods in one cycle – in this way a 9-bit conversion can be finished within 16 clock periods – typically 200ns.

We do not have the "result-cell". Instead of this we have two sets of cell 1 and 2. The steps 1 and 2 are equal as the steps 1 and 2 in the three-step algorithm described above. In step 3 (which starts new cycle) the currents from cells 1A and 2A are copied into 1B. In step 4 (corresponds to 2) 1A and 2A are copied into 2B.

The algorithm is illustrated in Figure 4.

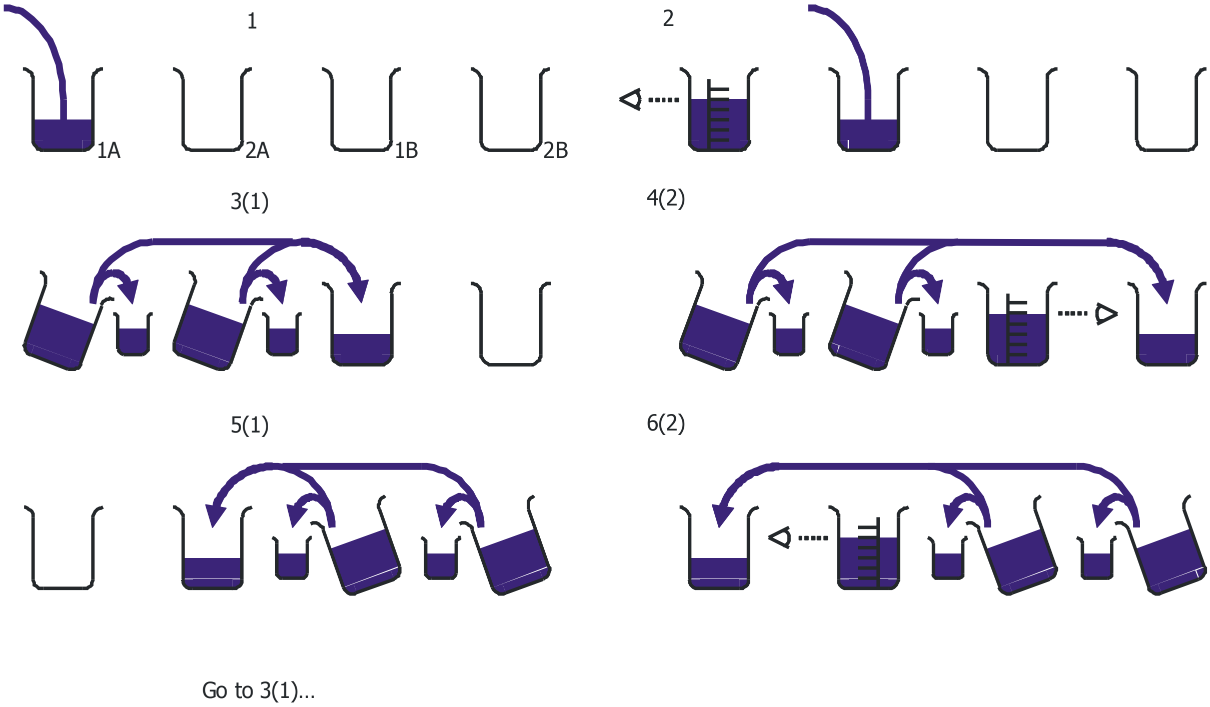

Figure 4: *Two step algorithm*

The big drawback is here that the reference currents (in my figure represented as small buckets) are used (added/subtracted) in every step. (In the three-step algorithm the reference currents are used only is step 3.) From this reason, there is no time to copy the reference current into some auxiliary current cell – i.e. to reverse its sign. We must use at least two different reference current sources, one current source with positive and one with negative sign. Matching of these current sources is an issue. Further, to simplify the schematics we actually have four reference current sources, two of them used when currents are copied from A to B and two of them for B to A. To improve the matching we have implemented the reference currents only with large PMOS transistors. If we need to add a positive reference, PMOS is connected to the output of the CMC. If we need to subtract a reference, another PMOS is disconnected from the output of the CMC.

The ADC as implemented on TCUM3 worked fine.

Pipelined ADC

We have also designed a pipeline version of the ADC. The working principle of the pipeline ADC is illustrated in Figure 5.

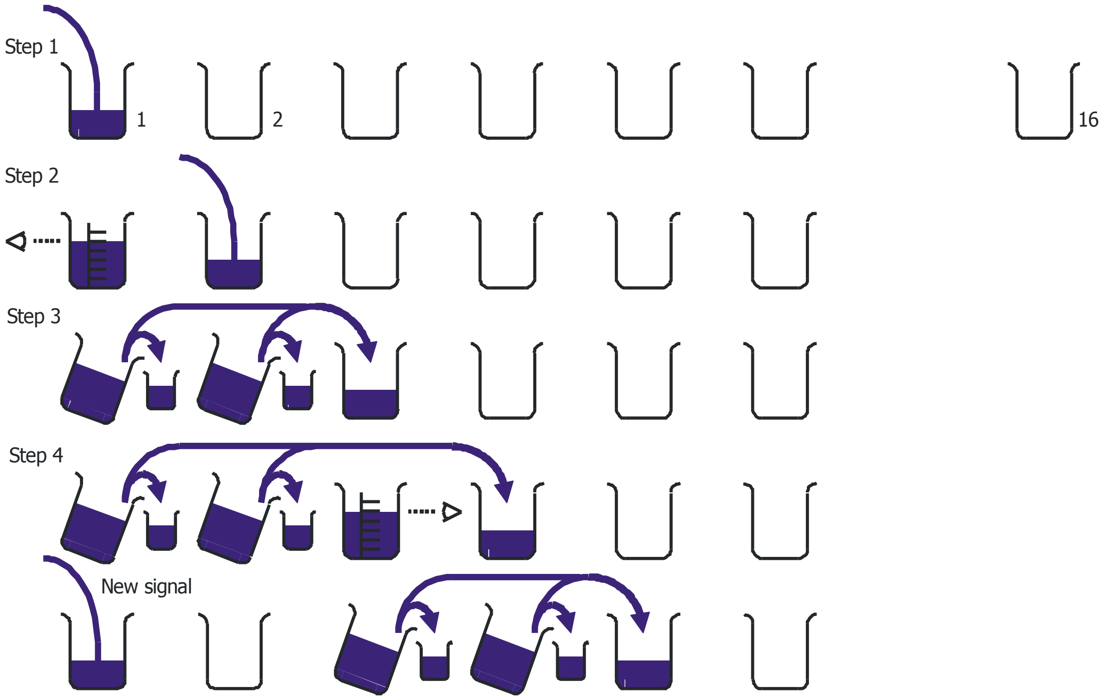

Figure 5: *Pipelined ADC*

Here we have for every cycle (we need 8 cycles) a set of two current memory cells. In total we have 16 CMCs. The conversion cycles can be then performed at the same time. In the case of our implementation, we need 4 clock periods (four steps) for one conversion. The conversion time is typically 50ns. We have increased the number of CMC cells (also layout area and power consumption) by factor of 4 and decreased the conversion time by the same factor.

We have implemented one additional improvement. The drawback of the concept from Figure 5 is that the input signal is sampled two times. It is quite obvious that it is equivalent: 1) to sample two times one signal "Sig" and 2) to sample once a signal that is two times stronger (2*Sig) into a current memory cell that is two times "larger" (made by connecting of two smaller cells in parallel). This is illustrated in Figure 6. Such a pipelined ADC worked fine.

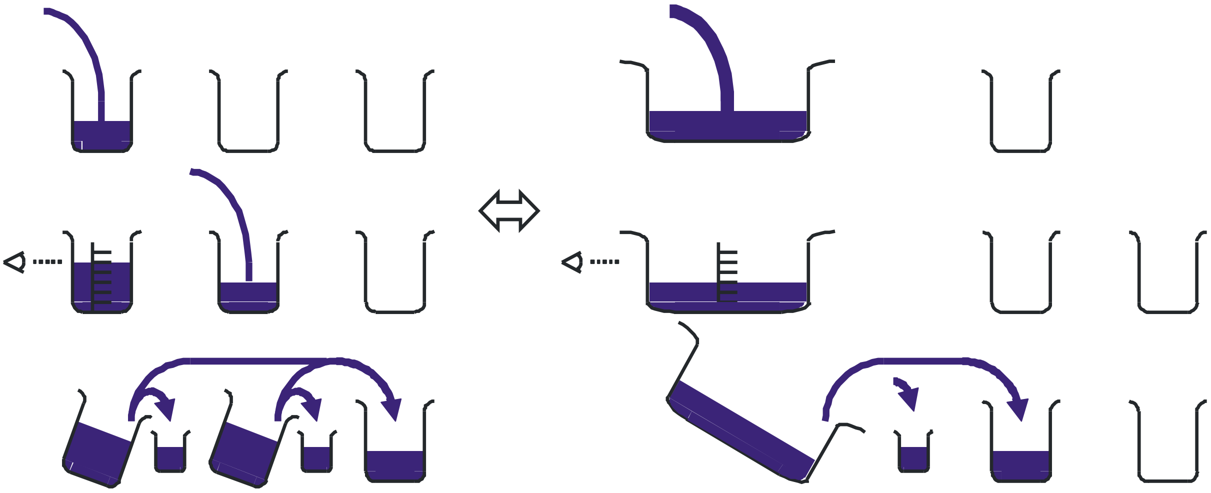

Figure 6: *Two equivalent ways of input signal sampling*

DCD1

DCD1 (2007) was the first test chip for the readout of DEPFET sensors based on the current-mode cyclic ADCs. The layout and the power consumption of the ADCs implemented on TCUM3 were too large. For DCD1, we have redesigned the cyclic ADC so that it works with much smaller bias currents (and has smaller power consumption). A new small layout has been done as well. The channel size was 180um x 110um. One channel contained two cyclic ADCs (8 current-memory cells in total). Additionally it contained a digital part that calculates the nine bit binary representation from the ADC bits ToHi and ToLo (contains bidirectional shift registers and a serial adder) and a DEPFET current receiver (regulated cascode).

There were 72 channels on the chip (144 ADCs in total). The data were transmitted out of the chip via 12 600Mbit/s LVDS differential outputs. Twelve ADCs (6 channels) shared one LVDS output.

The chip had bump-bond pads and it was possible to wire-bond it as well. For this purpose we had the wire-bond pads on the chip edges.

The chip worked but the noise was too high. The noise was probably the result of the following effect: The current memory cells (and the comparators) were done in the way that the power supply current depends on the currents stored in the cells. Since the layout was very dense, the power supply metal lines had relatively high resistances (~10 Ohm). The signal dependent changes of the supply current caused, in this way, oscillations of the power supply voltages and this caused the noise (crosstalk).

We have not observed such effects before because our previous chips had only several ADCs and the power supply lines were wider.

Figure 7 shows the schematics of the comparator and the attached “copy-transconductor” (“copy-TC”, see the first section) and it illustrates the problem of signal dependent supply currents. Especially critical is the AmpLow current since AmpLow is the reference power supply (source node) of the amplifiers. Notice that we are using single ended circuits (amplifiers, trans-conductors with one output). Fully differential circuits have been designed as well, but we have chosen the single ended variant due to its simplicity and smaller power consumption for equal speed. It would not be possible to make equally small layout in the case of differential circuits.

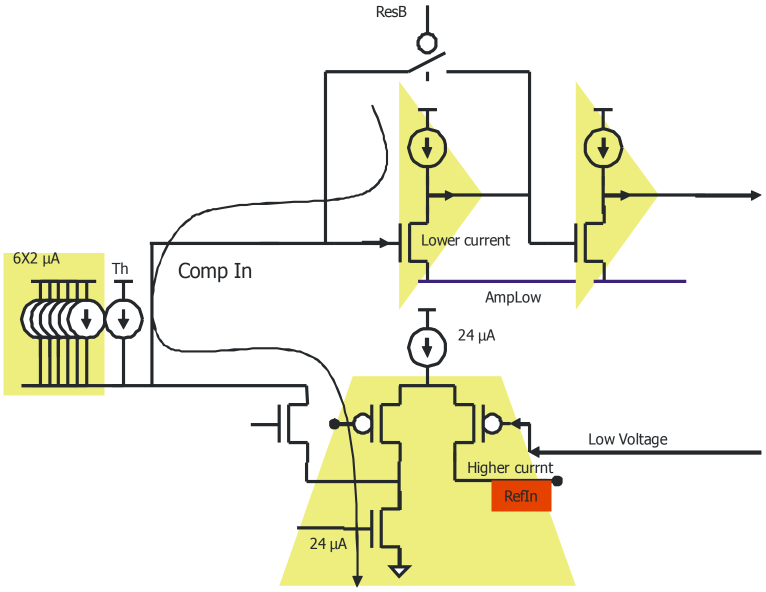

Figure 7: *Comparator and transconductor in DCD1*

DCD2

DCD2 is a modified version of the DCD1; it has been designed in 2008. The aim of the modifications was to make the current consumption signal independent and in this way remove the crosstalk. The modifications are illustrated in Figure 8. It shows the schematics of the comparator and the attached copy-transconductor.

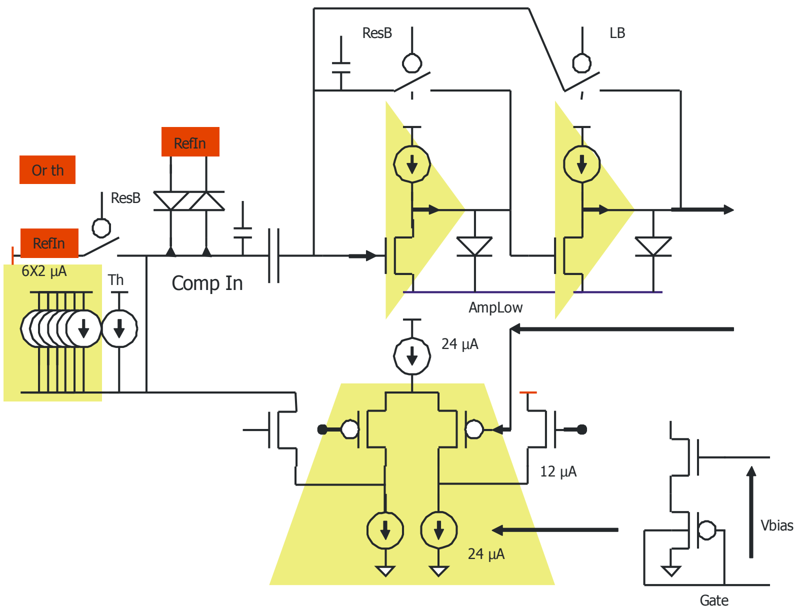

Figure 8: *Comparator and transconductor in DCD2*

It can be shown that the AmpLow and RefIn currents are now constant. The comparator is made faster by adding the positive feedback. DCD2 worked significantly better. It was tested in Heidelberg and Bonn (PhD work of Manuel Koch). The ADCs worked "perfectly" at lower clock frequencies. At high frequencies such as the Belle II nominal frequency (12.5ns clock period, 100ns sampling period for one channel - two ADCs) Manuel Koch measured INL (nonlinearity) of up to 4.5 LSBs (average 3.8) for 1.8V supply and 3 LSB (average 2.5) for 1.95V power supply (page 61 in his thesis). The input referred noise was 54nA in average. We have observed a clear frequency dependent increase of noise and INL. The INL increase was caused by long (missing) codes that emerged at higher clock rates.

At that time we have assumed that the current memory cells and the comparators were not fast enough. At high clock speeds the comparator may not work properly so that the thresholds moved more than allowed (12.5% of the range). Also the errors when currents are stored may be high. Nevertheless the achieved performances at 100ns sampling period were good enough.

DCDB1

DCDB1 was the first Belle II size chip (DCDB2 and DCDB4 have the same size as DCDB1). The analog part of the ADC was the same as in DCD2. This was the only part without changes. The DCDB1 chip is much larger than DCD2. It contains 256 channels and 512 ADCs in total. The channel size is 200um x 180um. Instead of the DEPFET current receiver in form of regulated cascode, we implemented a transimpedance (TIA) amplifier. The output of the amplifier is converted to current using a resistor. The use of TIA has several advantages over the regulated cascade. The time constant of the amplifier is signal independent. It is possible to implement gain by varying the resistor values.

The digital part that does the calculation of the binary numbers out of the ADC signals (TooHi/TooLow) is not placed in the channel. We have instead a large common digital block (occupies about 25% of the chip) which is shared by all channels. This block has 1024 digital inputs coming from the ADCs (TooHi/TooLow). 32 channels (64 ADCs) send their data via one 8-bit parallel output that operates at 320MHz. There are eight 8-bit outputs on the chip. The outputs are connected to low-voltage single-ended IOs. Such signals can be received by DHP.

DCD2 was designed to work in double sampling mode, in similar way as CURO.

The main operation mode of DCDB1 is single sampling. In this way, we can reduce the noise – the transimpedance amplifier has two times more time to amplify the input current. There is also one two-bit DAC in every channel that can be used to correct the pedestals before the TIA. The bits for these DACs are loaded from DHP before every new conversion.

Maybe the most significant change is that the DCDB chips are produced with bumps and with an extra aluminum redistribution layer. The production of these layers is not done by the foundry (UMC). The wafers are sent to another company. From this reason the production takes usually 6 months. The geometry of DCDBs is determined by the block size for the Europractice MPW runs. We use 6 blocks, the size of the chip is about 3.6mm x 5mm. Therefore, we do not have the freedom to choose any chip size we would like.

DCDB1 worked but it had a higher noise and worse INL at nominal clock speed than DCD2. The noise was 120nA at 100ns sampling speed. At lower clock speed, the results were better. We have found one small bug; a bias voltage that controls the delay of the CMC- sampling switch was a bit too low, leading to a large delay. Also, we have observed that some percentage of the ADCs did not work – the ADC characteristics were "broken" in one part of the input signal range. This was most probably caused by broken connections between the ADCs and the large digital part. These connections are long (~3mm) and the input inverters were most probably damaged due to antenna effect.

There is one bad effect of the use of chips with bumps. The time between the submission and the first reliable results is always long. First the production takes six months. Then, since we are using non-standard IOs (low voltage single ended) we cannot connect the chip directly to FPGA. We have designed a special converter chip (DCDRO) that converts the single-ended into differential signals. Both chips DCDB and DCDRO need to be flipped onto a silicon adapter. This adapter can be then wire bonded onto a PCB. The whole procedure is complicated and it takes time.

DCDB2

For DCDB2 we have done the following changes. Antenna diodes have been added and the bias voltage has been adjusted. The comparator has been slightly improved by adding a larger coupling capacitor and by connecting the negative clamp diode to a separate voltage VNMOS that is smaller than RefIn. The circuit is shown in Figure 9.

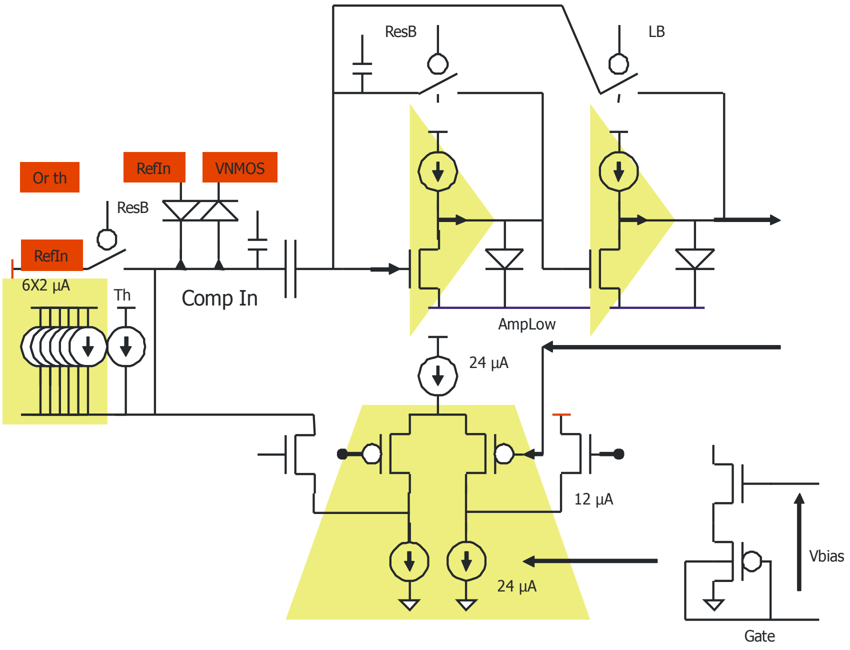

Figure 9: *Comparator and transconductor in DCDB2*

We have also implemented the analog common mode correction circuit. This circuit calculates the mean value of all DCD inputs (coming from DEPFET) and biases accordingly the subtraction current source connected to the TIA input. The time constant of the circuit is very short (~10ns).

DCDB2 works better than DCDB1. No broken ADC channels have been found which means that the antenna diodes helped. The noise at nominal speed (100ns) is about 65nA and the maximal INL 4.1LSB (mean value 2.6LSB). These values are similar as with DCD2. The result of the chip characterization measurement for one DCDB2 chip is shown in Figure 10.

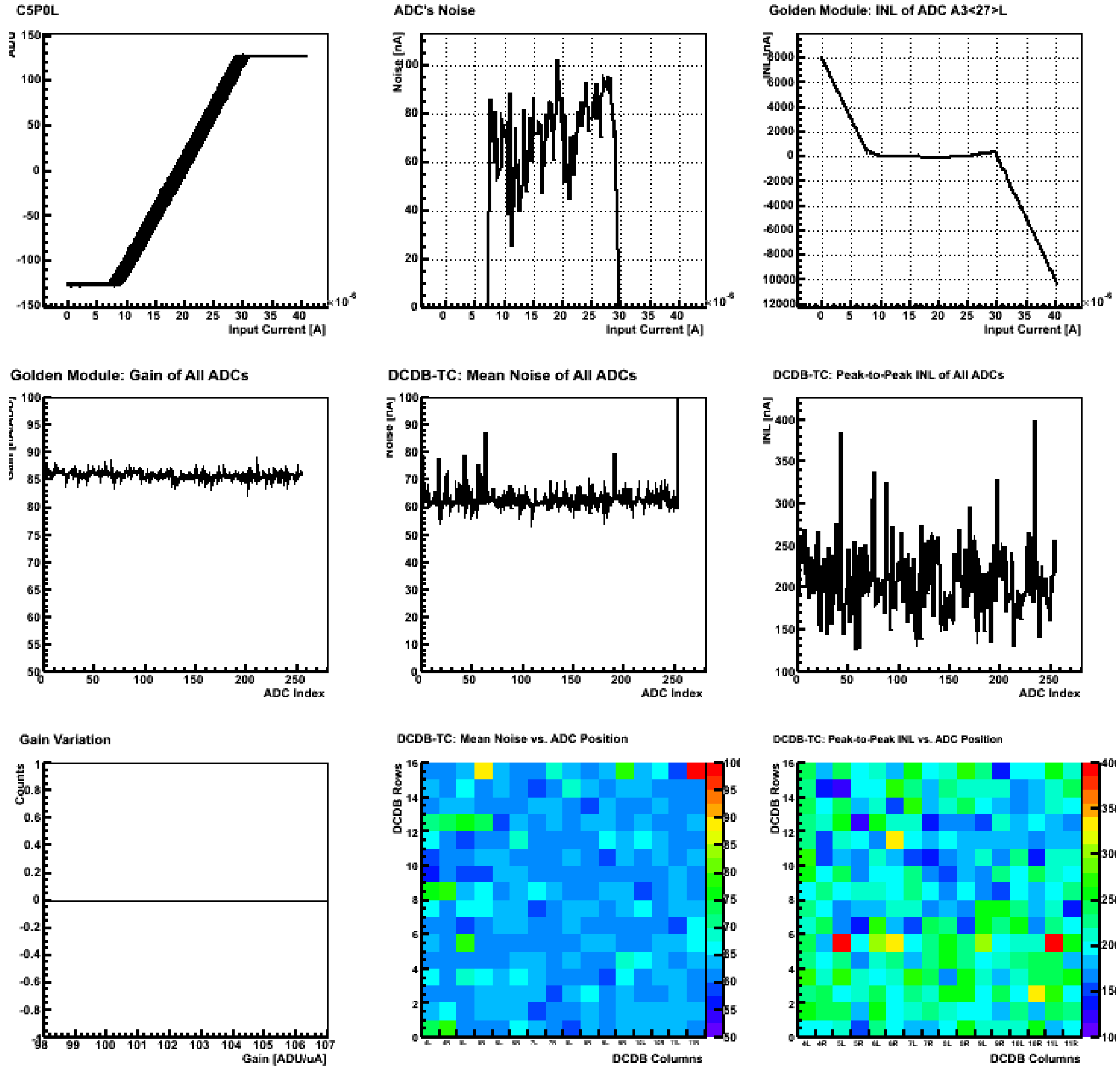

Figure 10: *DCDB2 characterization at 320MHz (100ns sampling period)*

The response speed of the trans-impedance amplifier has been measured on a test chip DCDB3 which has only 16 channels that are identical as these on DCDB2. This chip has been mounted onto the modular PCB system. The TIA inputs are connected to about 5cm long lines on the PCB. These lines emulate the DEPFET capacitance. The response speed has been measured by changing the sampling moment with respect to the signal injection pulse. Figure 11 shows the result.

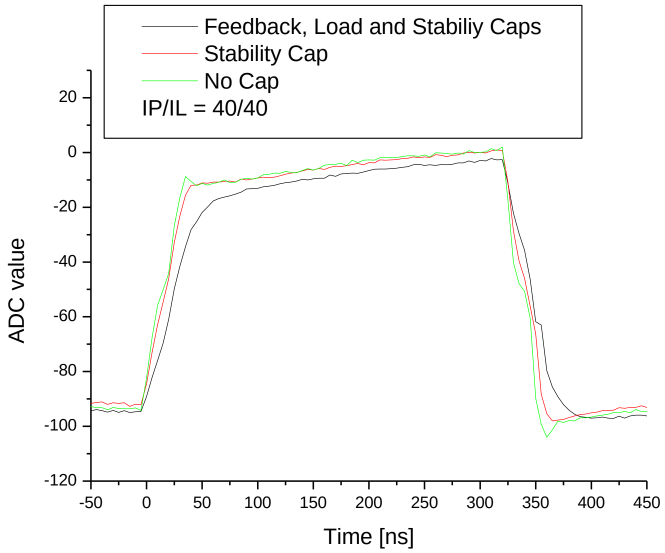

Figure 11: *DCDB3 test chip: response speed of the TIA*

There is a strange effect that the top is not flat. It is probably the influence of the test current source to the supply voltage. (We should repeat these measurements.) The rise time is fast enough (< 50ns).

Figure 12 shows the response of the transimpedance amplifier with the analog common mode correction turned on. The common mode correction is working. The amplifier is slightly slower in this operation mode.

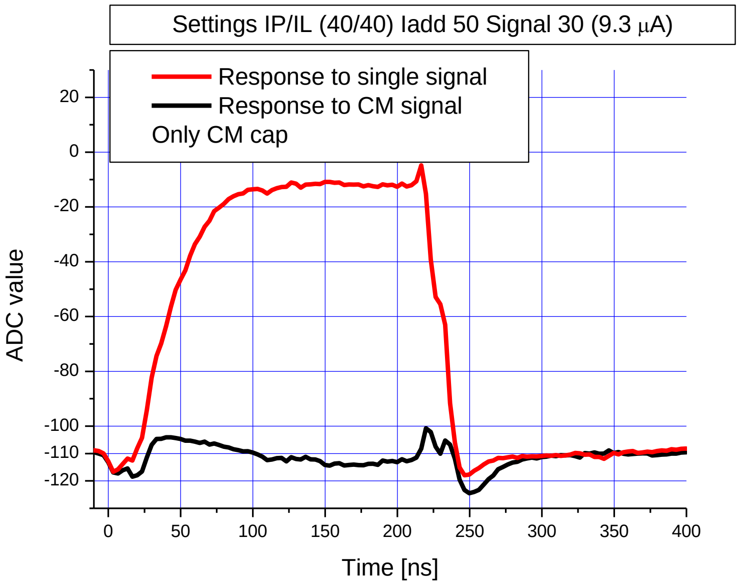

Figure 12: *DCDB3 test chip: response of the TIA with common mode correction turned on to a single signal of 9.3uA and to a common mode signal of the same value.*

The main (only to my knowledge) problem in DCDB2 is that some number of channels (typically 5%) has a bit higher INL (in the measurement above up to 4.1LSB). As in the case of DCD2, these ADCs show long missing codes. The situation is worse for higher clock frequencies. We have observed that increasing of RefIn helps. Our assumption was that in some channels the comparators are not working properly at high clock rates. The comparators are too slow. This assumption was in agreement with simulations of the ADC with large threshold offsets. The simulated characteristics are very similar to the observed ones. One simulated characteristics is shown in Figure 13.

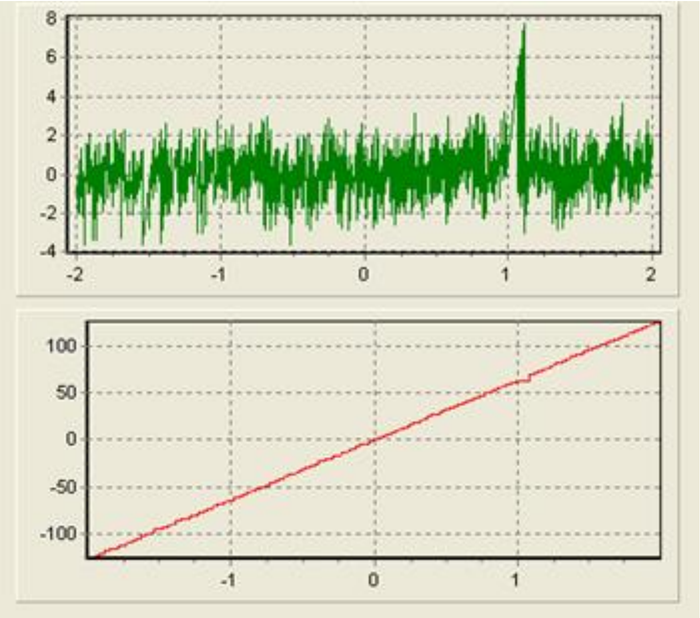

Figure 13: *Simulated ADC characteristics with large comparator offset*

We were looking for an explanation that only 5% of the ADCs are affected. We have concentrated our attention on the “dynamic” parts of the circuit – i.e. the parts in the comparator itself that transmit fast signals. Some ideas were that the coupling capacitors may be damaged (missing contacts, antenna damage).

For the new chip we have decided to go two ways. We have designed two improved versions one with the cyclic ADCs and one with the pipelined ADCs.

DCDB4_cyclic

The DCDB4_cyclic is very similar to DCDB2. We have slightly optimized the layout of the ADC and of the comparator (added antenna diodes and doubled contacts). The only change in the schematic of the comparator is that the positive clamp diode has a dedicated higher voltage VPMOS (Figure 14). This change was motivated by the fact that higher RefIn values improve the characteristics.

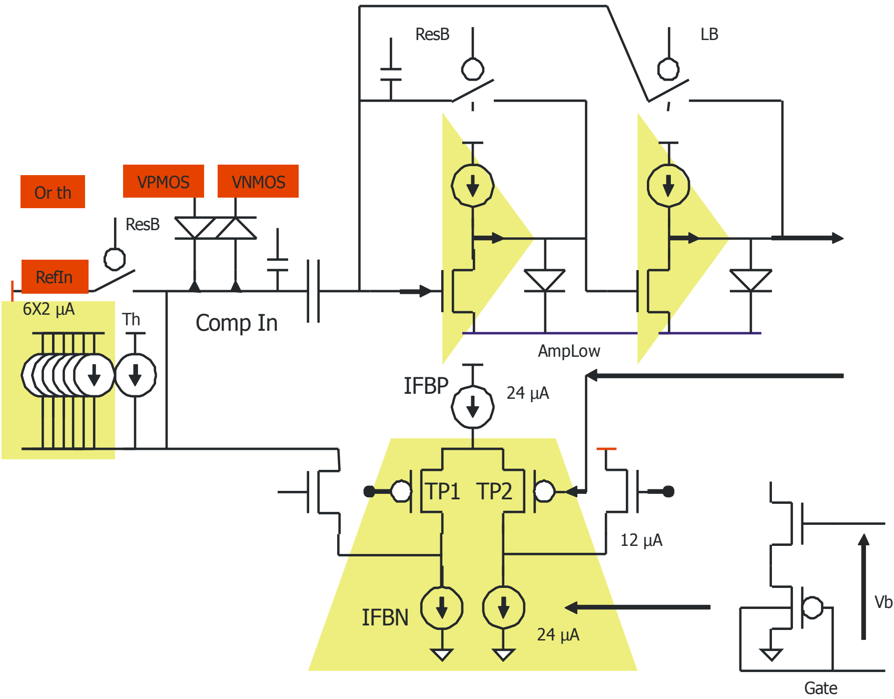

Figure 14: *Comparator and transconductor in DCDB4*

Further changes are: 1) Improved on-chip test current source with the on-chip DAC. The DAC LSB corresponds roughly to the LSB of the ADCs. This allows detection of long missing codes using only on chip-test current source. 2) Temperature stabile bang gap current reference. 3) Analog common mode correction can be switched on/off on channel level.

So far we did not have time to test any of DCDB4_cyclic chips.

DCDB4_pipeline

DCDB4_pipeline uses one pipeline ADC instead of two cyclic ADCs. The pipeline ADC is two times larger (16 CMC cells) than the two cyclic ADCs together (4 cells each). Assuming the same clock frequency, the pipeline ADC is two times faster. Our motivation was, however, to relax by factor of two the clock period so that we achieve the same sampling period (100ns) by clocking the current memory cells twice slower (25ns clock). This should be compared with 12.5ns clock in the case of the cyclic ADC. We wanted in this way to make the comparators working properly even at maximum speed and to eliminate the missing code problem. A completely new digital block had to be designed for the pipeline ADC.

DCDB4_pipeline is working fine however the missing (long) code problem did not disappeared completely. It is possible to adjust the chip so that the most of the channels are working nice. The result of the DCDB4_pipeline characterization with optimized settings is shown in Figure 15.

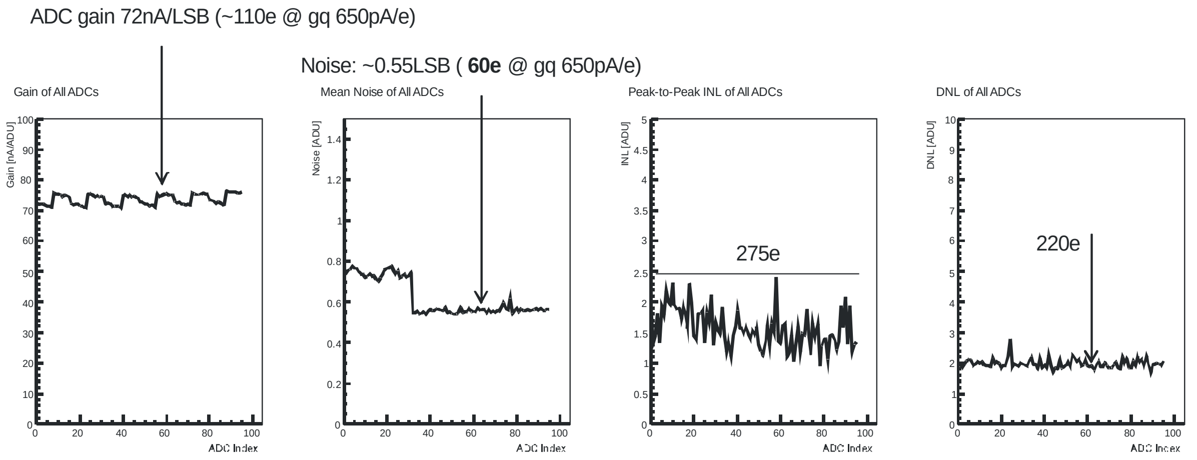

Figure 15: *DCDB4_pipeline characterization at 320MHz (100ns sampling period)*

All of the channels have INL lower than 2.5 LSB (peak to peak value) and the noise is 45nA. These are the best results so far. However, this is the best chip and the best optimization we had so far. If the chip is not well optimized much more channels have the missing codes (about 5%). It seems that we did not identify the origin of the missing cods correctly. We suspected a slow comparator, broken capacitor or too low clamp voltage. We have reduced the clock speed (which helps if the comparator is too slow), improved the layout of the capacitor and changed the clamp voltage but the problem is still there.

There is one very simple explanation of the problem: the mismatch between the original and the copy transconductor. This is illustrated in Figure 16.

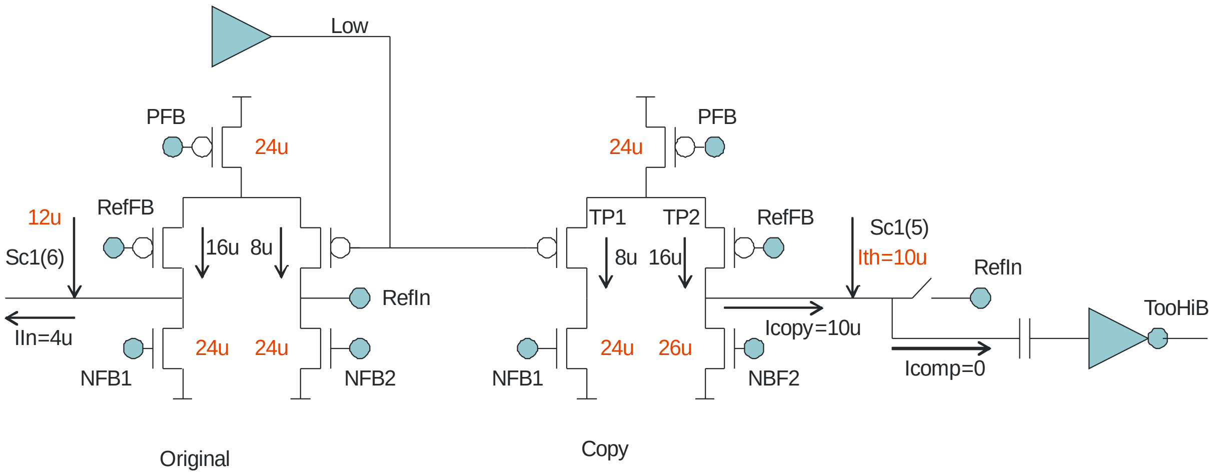

Figure 16: *Mismatch between the transconductors*

The right transistor NFB2 in the "TC-copy" has a bias current of 26uA, instead of nominal 24uA. The error in the copy of the input current (Icopy) is 2uA (10uA instead of 12uA), which is 12.5% of the signal range. This means, due to the complex design of the transconductor, a mismatch of only 2/24 = 8.3% in one device can cause a dangerous change of the threshold voltage. When the ADC is simulated under such conditions, we obtain the result as in Figure 13. This fits very well to the measurement results. Notice that the devices PFB, TP1 and TP2 contribute to matching too. It is very probable that the RMS square of the four mismatch contributions exceeds 8.3%. One other fact speaks in favor of the mismatch theory. If we measure a long code around 64, as in the simulation from Figure 13, *decreasing* of RefIn helps. According to our theory and simulation, a long code around 64 is caused by too strong NFB2 transistor (if generates too high current). If we decrease RefIn, we act against the mismatch since NFB2 has a finite output resistance.

On the other hand, if we measure a missing code around -64 *increasing* of RefIn helps. This situation can be reproduced is we assume that NFB2 is too weak.

We have probably exactly this problem of mismatched transistors from the beginning of our ADC development. However, the transistor mismatch was in many designs hidden by an additional mismatch due to the slow comparator and fast clocking. The both contributions are superimposed. For instance, in the case of Iin = 4uA in Figure 16 we should obtain a stabile TooHi result. Due to mismatch in NFB2 (too high bias current) the comparator is on the edge (Icomp=0 instad of Icomp=2uA). If we assume that the comparator is additionally affected by fast clocking we will obtain a wrong result of comparison and the missing codes around 64. We will verify our theory by measurements of the transistor currents. It is possible to access the output of one transconductor in every channel from outside. In this way we can check which transistor has the largest mismatch.

The fix in the next chip will be to resize the critical transistor and in this way improve the matching. We should notice that devices NFB have a complex structure from Figure 14. The devices contain two NMOSTs and one PMOST. We use this structure since DCD2 to make the current sources less susceptible to changes in the local GND level. The two NMOS transistors have enclosed gates. All this may influence matching and we will redo the layout carefully. Figure 17 shows the layout of the transconductors.

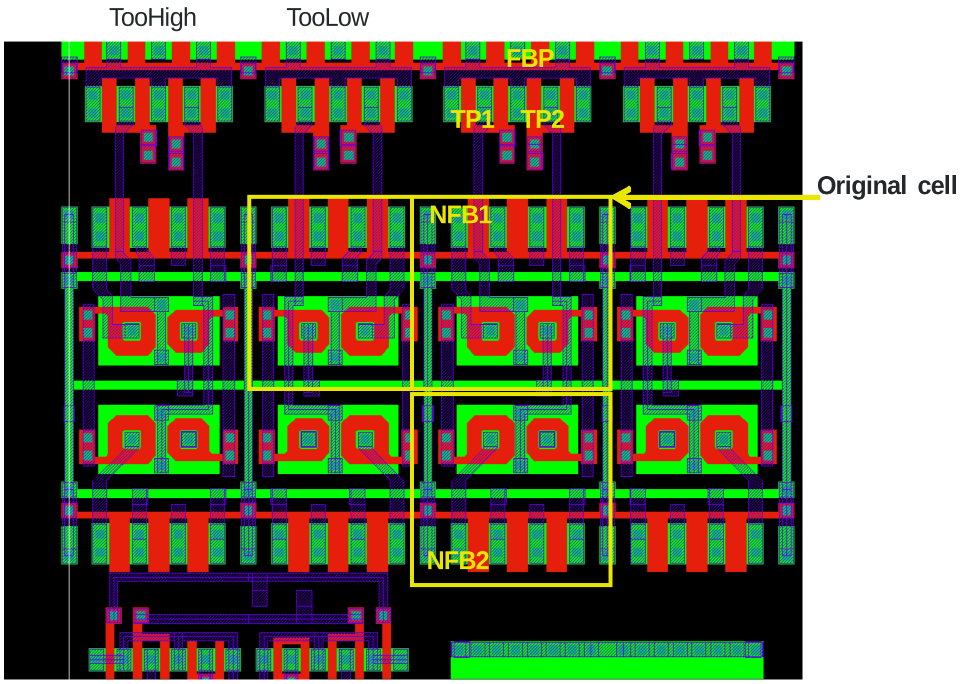

Figure 17: *Layout of the transconductors.*