DCD

Simulations of Digital Output Driver, ADCs, Monte Carlo Simulations

Ivan Peric

Digital transmission from DCD to DHP

During the July review several possible output driver schemes have been presented (pages 7-11). It seems that the present driver scheme, with or without increased bias current, is the best choice. In the following section I present the simulations of this driver, where I mostly concentrated on the scheme with the same bias current as now. Following recommendations of my colleagues I would not change the transistor layout to enclosed one.

Present status

Fig 1 shows the simulation of the data line waveform with the present DCD output buffer, for the line capacitance 1-3pF (typical capacitances) and for the present DHP input capacitance of 3pF (see the DHP presentation from the review in July).

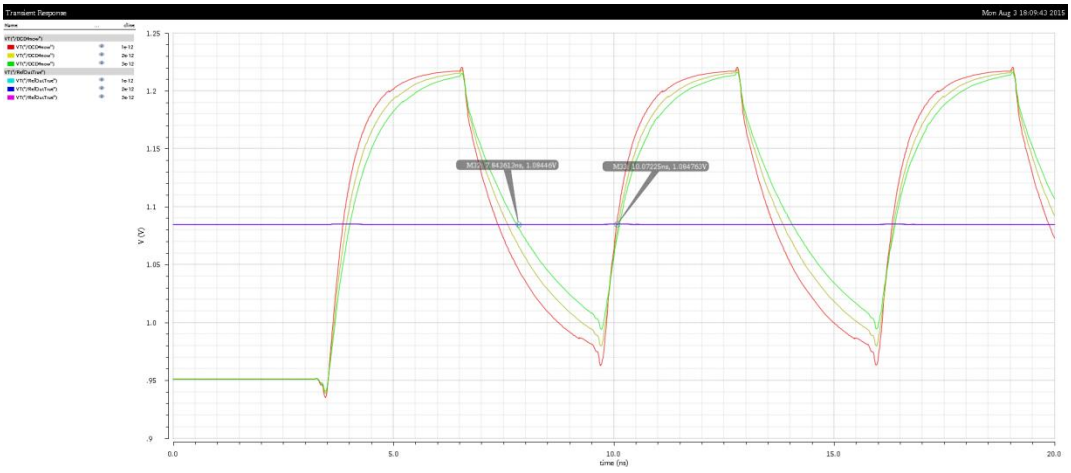

Figure 1: Data line waveform.

The reference voltage - RefOut, generated by DCD is also shown.

There is a duty cycle asymmetry; however the width of the logic zero is 2.28ns. (Ideally it would be 3.125ns). 2.28ns should be long enough for a safe sampling of logic zero. However, our measurements show much smaller sampling window.

The following simulations show that the main problem in the present system is that the DHP receiver has a hysteresis.

Hysteresis means that the DHP receiver (in logic-0 output state) generates logic one if the signal is higher than RefOut + Vrise. The DHP receiver (in logic-1 output state) generates logic zero if the signal is lower than RefOut – Vfall. From page 55 in DHP presentation we find the following values: Vrise = 57mV, Vfall = 70mV.

I have done following simulations using a generic differential receiver, which has similar speed as the DHP receiver. I have added hysteresis using ideal components from the Cadence component library ("vcvs"). The hysteresis can be varied.

Figure 2 shows the output of the generic differential receiver (GDR) when it receives the signal that we have in the present system (Figure 1) and when it has a hysteresis as DHP now (Vrise = 57mV, Vfall = 70mV). We see that the receiver fails to detect logic 0 for the line capacitance of 3pF. (For clarity, Figure 3 shows only the waveforms for the case of the line capacitance 3pF.)

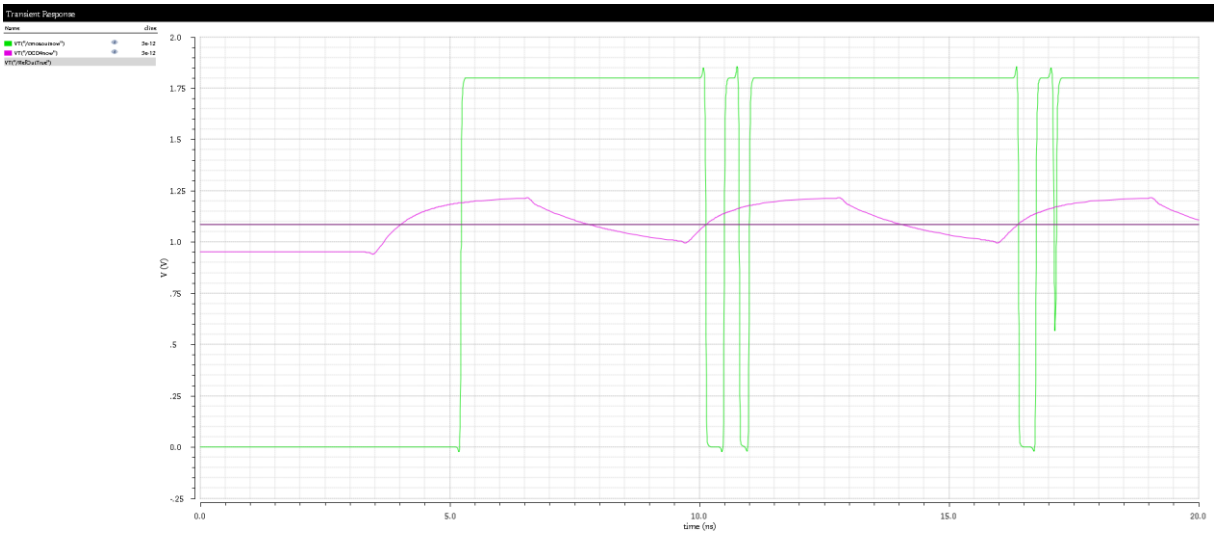

Figure 2: Simulation of the present data communication – generic receiver with hysteresis.

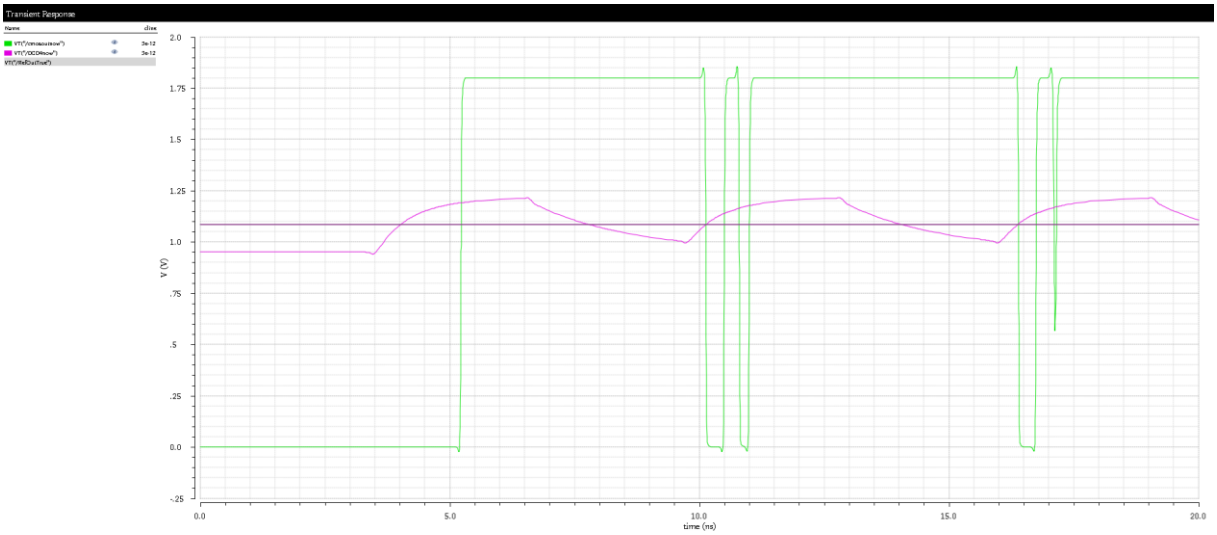

Figure 3: Simulation of the present data communication – generic receiver with hysteresis. Result for only 3pF line capacitance.

Such a simulation result can explain our observed problems. The hysteresis is too large for the signal we have.

In DHP presentation a similar simulation has been shown in page 54, where the effect does not look so severe. However, the waveform used for this DHP simulation is different than the real one (from Figure 1). The time constants of the waveform from the DHP simulations are much smaller than I simulate in Figure 1. As crosscheck, Figure 4 shows the simulation of the generic differential receiver with the waveform from the DHP presentation. The output waveform is similar as in the DHP slides, which shows that the generic differential receiver with hysteresis behaves similarly as the DHP receiver.

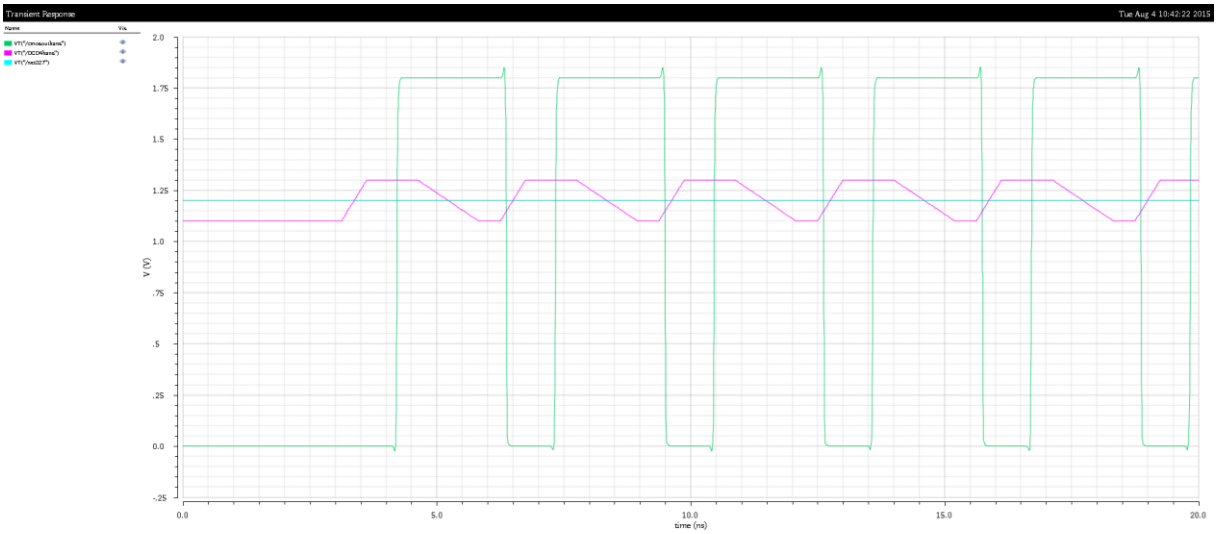

Figure 4: Generic receiver with hysteresis receives the waveform from the DHP presentation.

In order to improve the digital communication DCD -> DHP I propose the following:

1. Remove the hysteresis from the DHP receiver.

Figure 5 shows the waveform after the generic differential receiver without hysteresis with the present signal as input (line capacitance 1-3pF, DHP input capacitance 3pF).

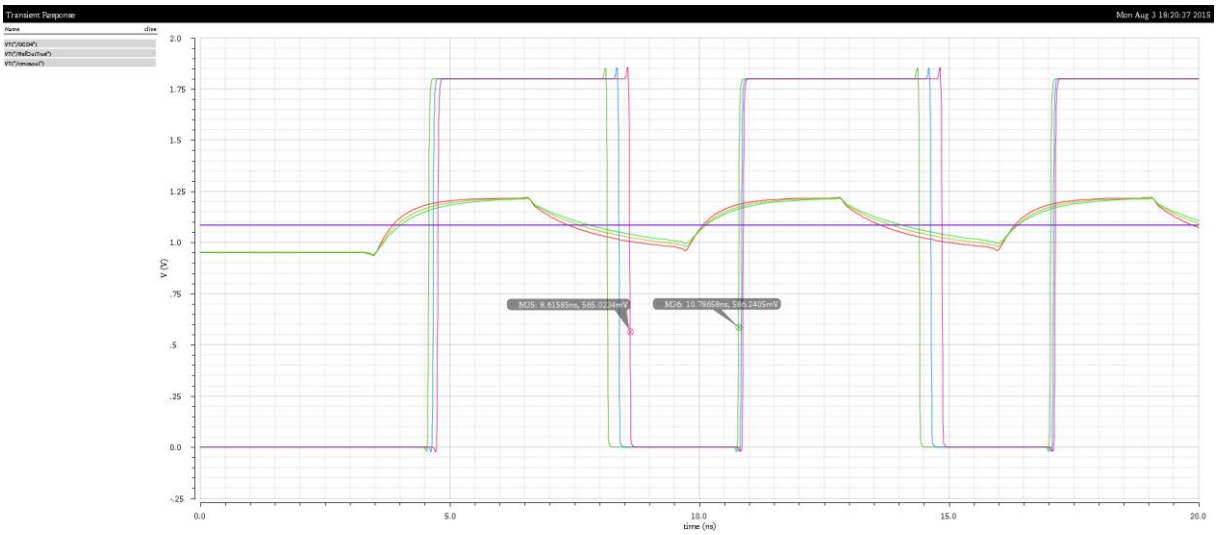

Figure 5: Generic receiver without hysteresis with the present signal as input.

The sampling window (measured at the CMOS signal) for logic zero is 2.171ns, which is similar as when we measure at the low voltage input (Fig 1).

(I "defined" the sampling window as the time between the signal falling edge for 3pF and signal rising edge for 1pF capacitance. This leads to a ~100ps smaller time window.)

Notice: The proposed DHP receiver has a small hysteresis (page 56 DHP presentation), which, to my opinion, should be reduced to zero.

2. Decrease the DHP input capacitance from 3pF to 1pF.

Figure 6 shows the corresponding waveform after the generic differential receiver without hysteresis with the line capacitance 1-3pF and the reduced DHP input capacitance of 1pF.

Sampling window for logic zero is now improved to 2.458ns.

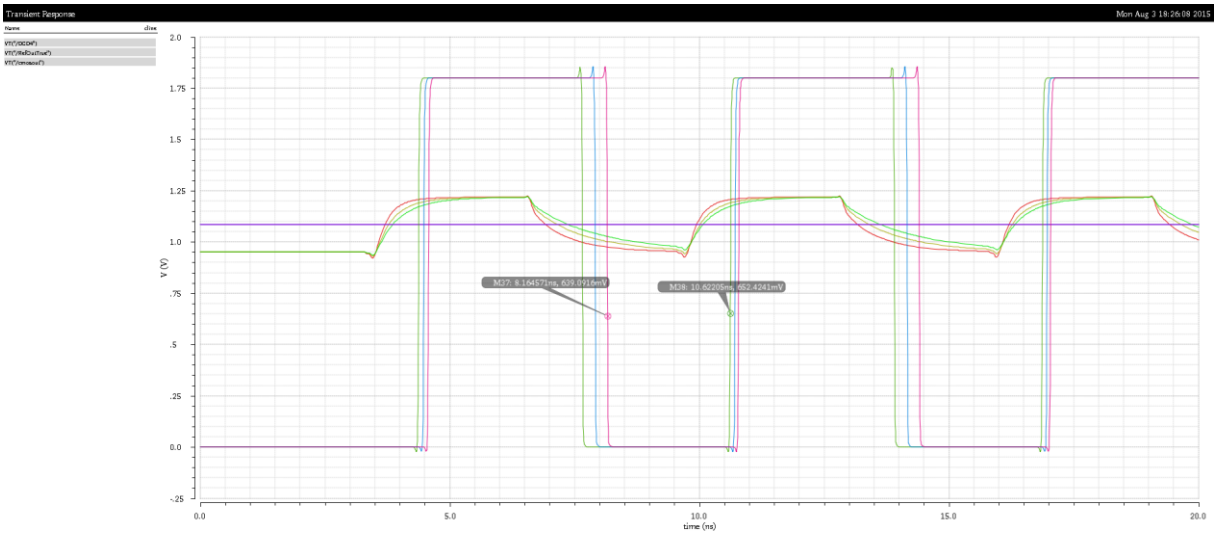

Figure 6: Generic receiver without hysteresis with the reduced DHP capacitance.

The decrease of the DHP input capacitance 3->1pF has been proposed in the DHP presentation.

At this place I show the simulation with the reduced DHP capacitance and the DHP receiver with small hysteresis that is proposed by DHP designers – Fugure7. The sampling window is 1.859ns, which is in principle fine, but 25% worse when the receiver without hysteresis is used.

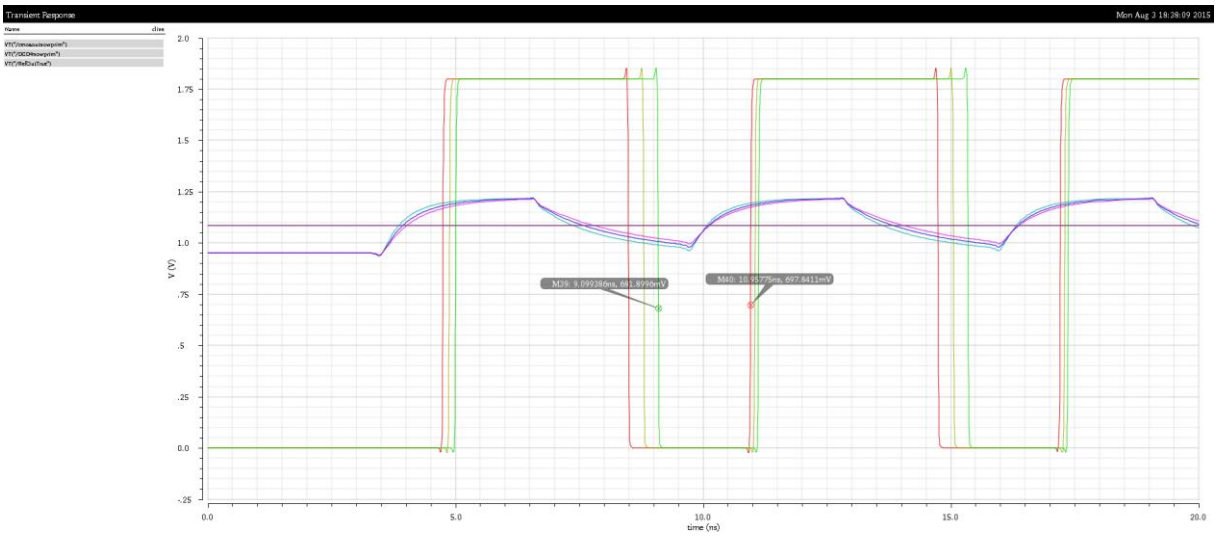

Figure 7: Generic receiver with hysteresis proposed in DHP document.

Notice, all the previous simulations have been done with the present DCD output driver. During our collaboration-internal discussions, my colleagues proposed to add a switch that can (optionally)

Increasing of the driver bias current is to my present knowledge not needed, however, adding of the additional switch and the resistor for bias boost (from 1.334mA to about 1.838mA) does not impose any danger and adds safety - it can be done.

Simulation in Figure 8 shows the waveform after the generic differential receiver with higher bias current (1.838mA) without hysteresis, with line capacitance 1-3pF and with the reduced DHP input capacitance of 1pF. The sampling window is now 2.45ns, a bit worse than with low current, however the difference between logic one and zero is larger.

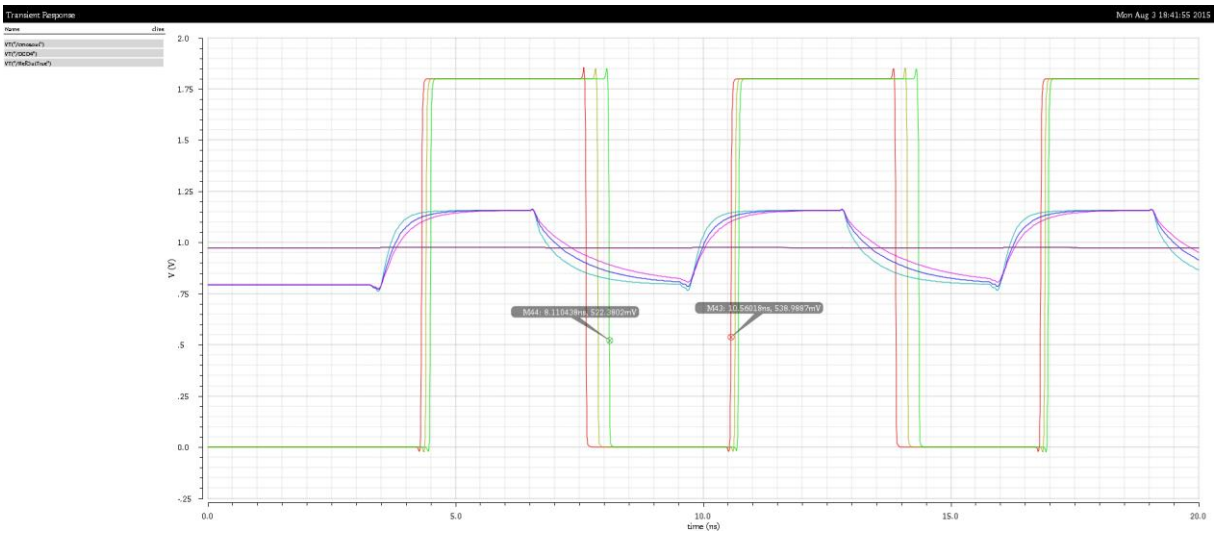

Figure 8: Driver with higher bias current.

In the following tables I give the overview of the logic zero sampling windows for different schemes:

Now (large hysteresis, large DHP cap., present DCD driver): 0!

DHP-proposed (small hysteresis, low DHP cap.), present DCD driver: 1.859ns

No DHP-receiver hysteresis, low DHP cap, present DCD driver: 2.458ns

No DHP receiver hysteresis, low DHP cap, DCD driver with increased current: 2.45ns

DHP-proposed, DCD driver with increased current: 1.96ns

From this table I propose to remove entirely the hysteresis from the DHP receiver, to reduce the DHP input capacitance from 3-1pF. For DCD, according to my simulations, the present scheme is fine. A possibility to increase the bias current by switching additional bias resistor in parallel can be added to increase the distance between logic one and zero.

I have additionally done following simulations of the present DCD driver with the generic receiver with no hysteresis, the DHP capacitance of 1pF, line capacitance of 3pF.

Corner analysis

Typical corner: sampling window: 2.6ns

Slow corner: sampling window: 2.7ns

Power supply rejection ration

In this simulation the vddd voltage has an additional AC- (pulsed-) component of 50mV (power supply noise)

Figure 9 shows 1. the analog driver-out waveform, 2. the CMOS waveform after the generic receiver and 3. the original clock. The “jitter” produced by the power supply noise is ~200ps or 7.7% of the sampling window. This is not a problem.

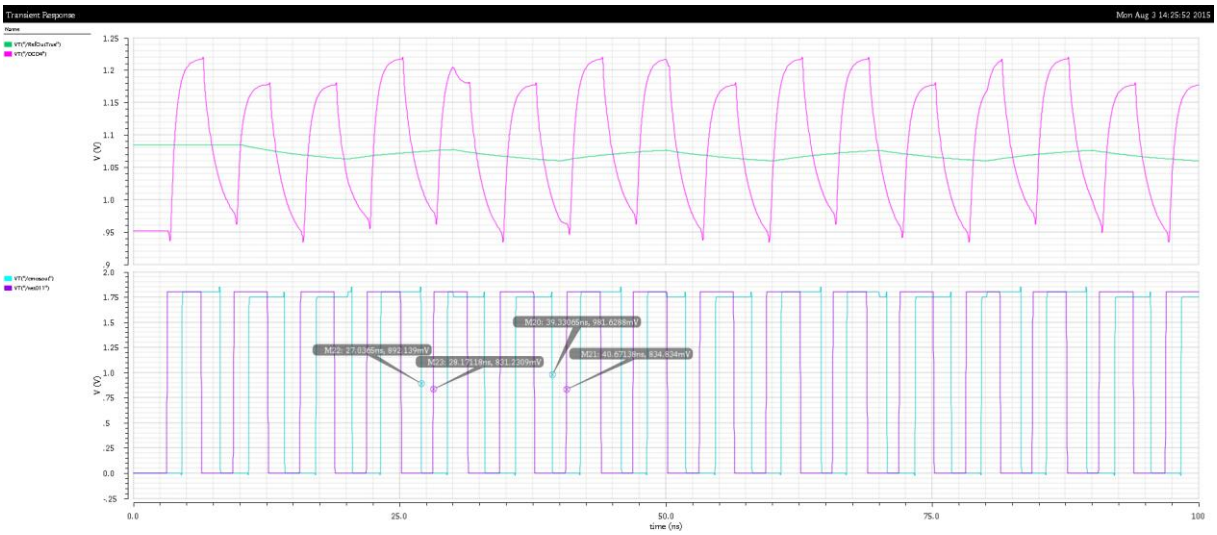

Figure 9. Simulation of power supply noise.

Figure 10 shows the same simulation as figure 9 with additional transient noise – no visible effect.

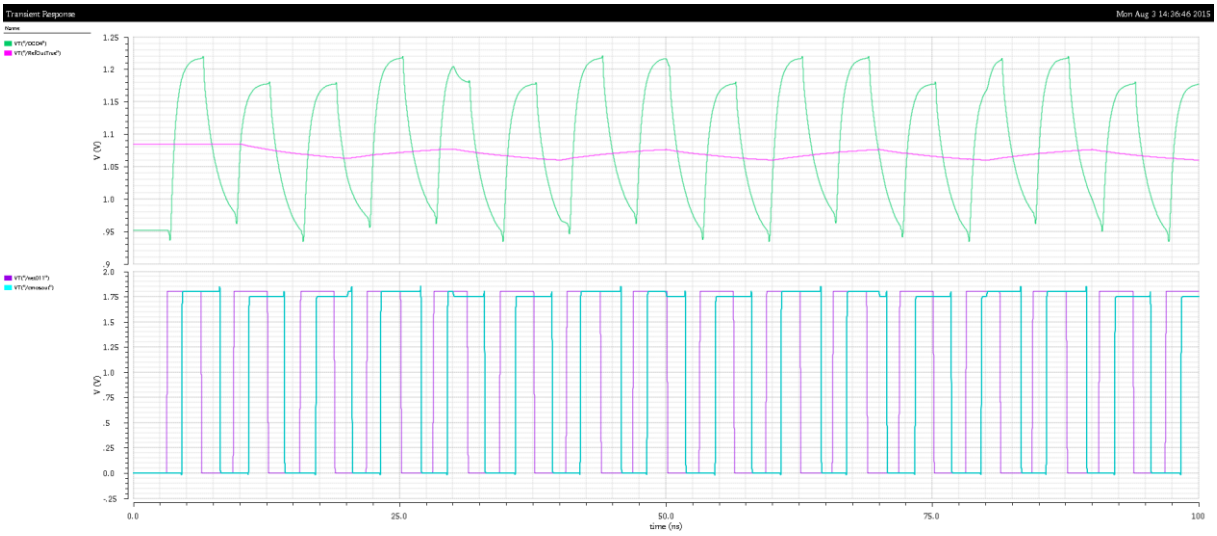

Figure 10. Simulation of power supply noise and transient noise.

Figure 11 shows the driver waveforms simulated from schematics (DCD) and simulated with QRC extracted layout (DCDQRC). The difference is negligible.

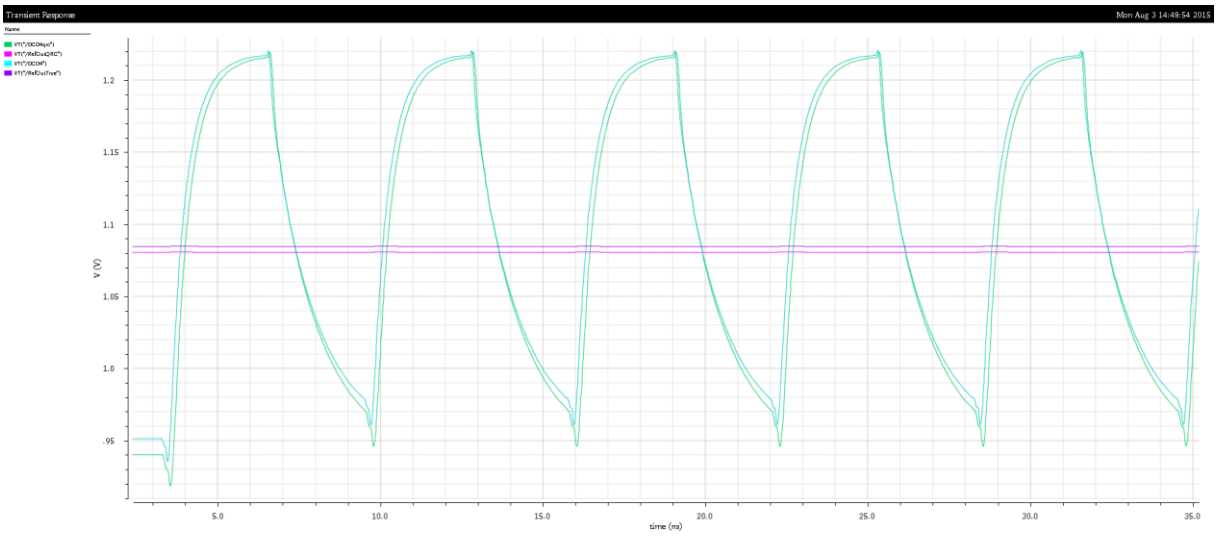

Figure 11. Simulation from extracted layout and from schematics - comparison.

I have also checked the power scheme and the resistances of the digital power lines.

The digital power is distributed from the bump bond pads (16 for vddd and 16 for gndd) via 20 vertical M5 power lines to the bottom of the chip. The 20 power lines are connected horizontally by 16 horizontal power lines in thick M6 metal.

The whole power mesh is connected to the horizontal thick M6 power lines that are used only for IOs (drivers are receivers).

The resistance of the 20 vertical power lines is about 0.5 Ohm (0.33 typical value).

The horizontal M6 line for the drivers has the total resistance of 2.25 Ohms. We expect a maximal voltage drop of about 48mV on the horizontal line for the standard DCD driver current.

Figures 12 and 13 show the influence of the voltage drop of 50mV to the difference between the logic 1 and RefOut and between the logic 0 and RefOut. The difference is reduced typically from 133 mV to 112mV for logic zero. This is not a problem; the sampling window is still ~2.6ns, which is ok.

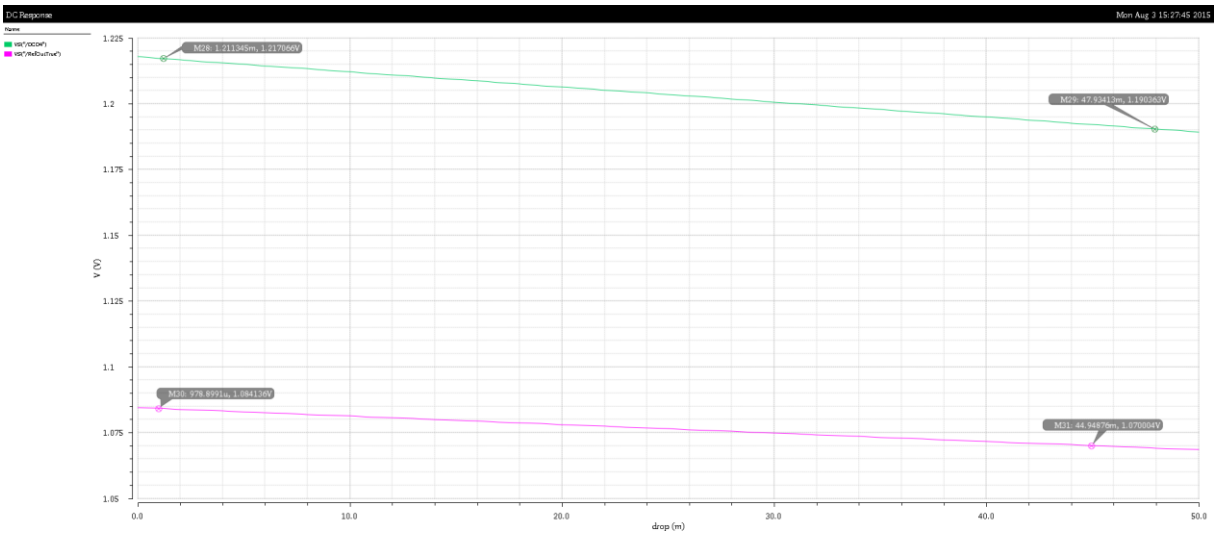

Figure 12 - Influence of the voltage drop of 50mV to logic 1-Ref difference.

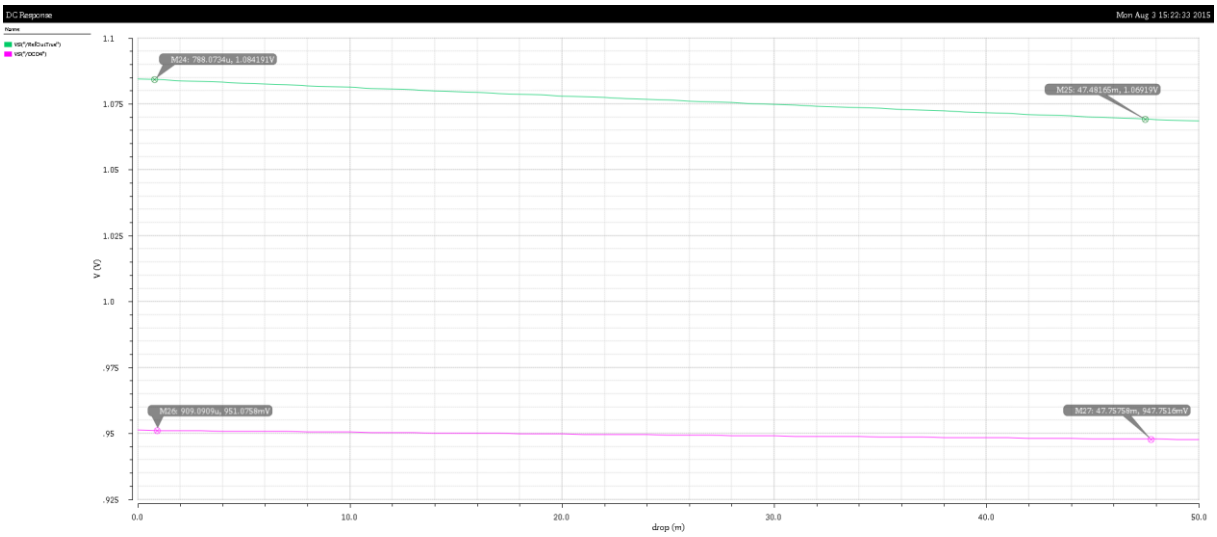

Figure 13 - Influence of the voltage drop of 50mV to logic 0-Ref difference.

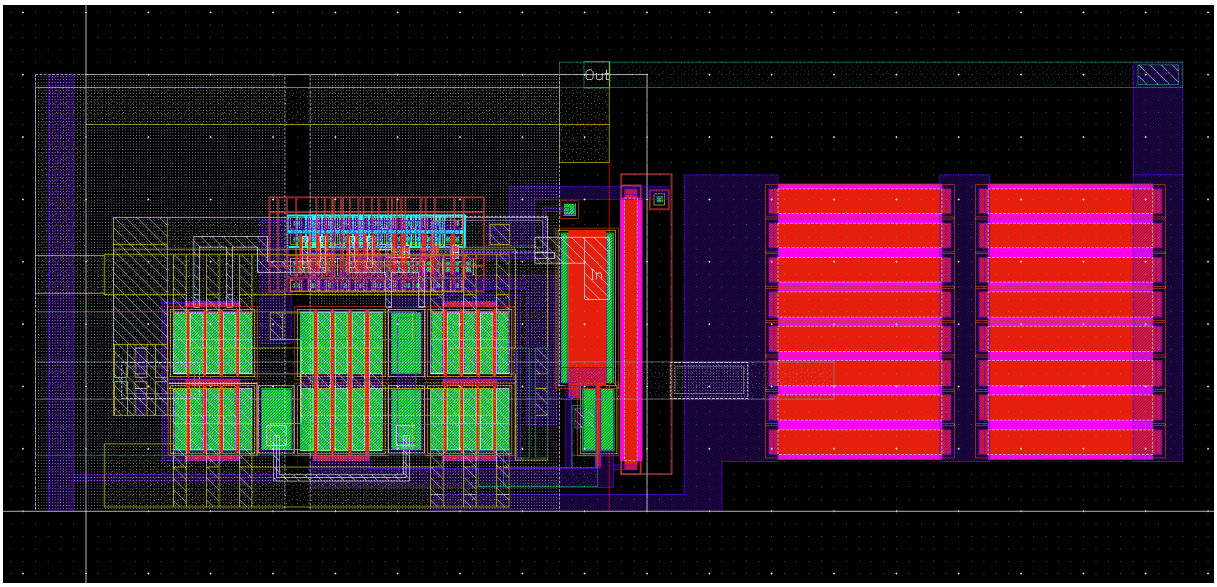

Figure 14 shows the layout of the output driver.

Conclusion: I propose to remove entirely the hysteresis from the DHP receiver, to reduce the DHP input capacitance from 3-1pF. For DCD, according to my simulations, the present scheme works well. A possibility to increase the bias current by switching additional bias resistor in parallel can be added to increase the distance between logic one and zero.

Notice that the DCDRO chip, designed in Heidelberg to be used as an interface between the DCD and FPGA, uses the same differential receivers (without hysteresis) that have been used in the simulations above. DCDRO and DCD chips are bump bonded onto the same silicon adapter chip and used on “hybrid 4” PCBs. Data communication between DCD and DCDRO works on hybrid 4 systems without errors, even without the possibility to fine-tune the delay for every channel.

Voltage drops on power lines on DCD

When we talk about voltage drops in DCD we can distinguish two types: global voltage drop from channel to channel and local voltage drop within one channel (or ADC). The global voltage drops would lead to poor functional ADCs at one side of the DCD-channel matrix. The local voltage drops could lead to mismatch within components of the ADC.

In this section I concentrate on global voltage drops.

During review I have presented several improvements (slides 12-16) that could reduce the global voltage drop. However, here I will first try to estimate whether the existing voltage drops are problematic.

The layout of the DCD channel is quite dense and complex and it is very difficult to find space for power lines. The main power lines are placed on the thick M6 metal layer showed in Figure 1.

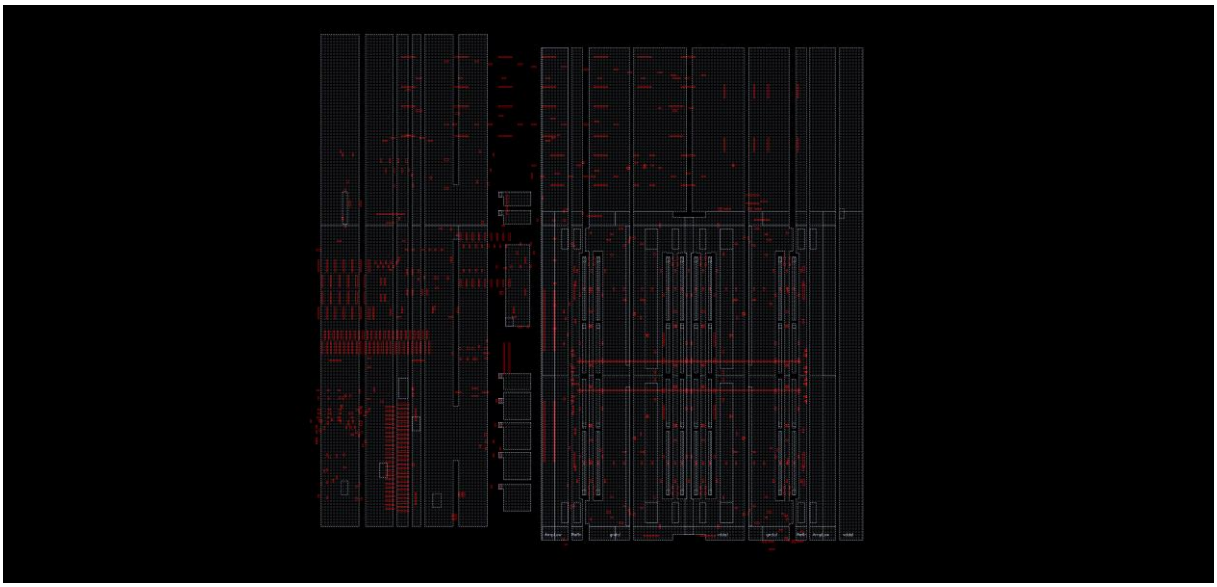

Figure 1: Layout of the top metal in DCD channel.

The 256 DCD channels are organized in 16 columns each 16 channels. For one column I have extracted the following resistances and voltage drops:

VDDA	Resistance: 2.634 Ohm (max)	Drop: 32mV (max)
AmpLow	Resistance: 4.547 Ohm (max)	Drop: 34.9mV (max)
RefIn	Resistance: 12.632 Ohm (max)	Drop: 19.7mV (max)
GNDA	Resistance: 3.512 Ohm (max)	Drop: 21.62mV (max)

These voltage drops can affect the following two important bias currents of the ADC – 1) PFB current (PMOS differential-pair bias of the trans-conductor), 2) NFB current (NMOS load of the trans-conductor).

I have simulated the effect of the above mentioned voltage drops to these two currents:

It should be mentioned that the voltage drop is leading to a *higher* current, since the bias block (containing the diode-connected transistors that convert DAC currents into voltages) is connected to the upper part of the DCD matrix, where the drip is the highest. This means the bias currents in the bottom most channels are higher.

NFB

The NFB current source uses the special circuit that is insensitive to voltage drops. Therefore I simulate:

I_NFB = 11.61uA (no voltage drop), I_NFB = 11.91uA (voltage drop)

PFB

PMOS current source uses the standard scheme (a transistor). Because of this, the voltage drop is higher. (It was not feasible to make the improved scheme on the PMOS side.)

I_PFB = 24.04uA (no voltage drop), I_NFB = 30.25uA (voltage drop)

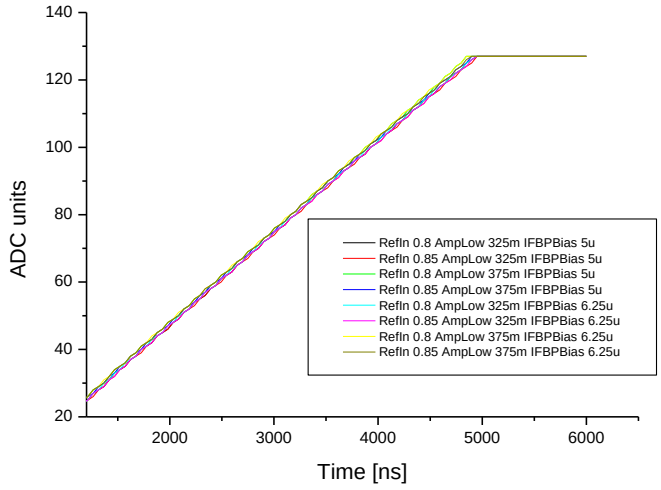

Figure 2: Transient simulation of ADC at doubled clock speed – different settings, slow transistor corner.

Figure 2 shows the "corner" mixed mode simulation of the full DCD channel connected to the DCD digital block. We have chosen the slow transistor corner, and we have varied the PFB current in the range 24uA to 30uA (DAC bias current 5uA-6.25uA), the AmpLow voltage +- 50mV from nominal value and RefIn voltage +-50mV from the nominal value to simulate voltage drops.

Figure 3 shows the same measurement for typical transistor corner.

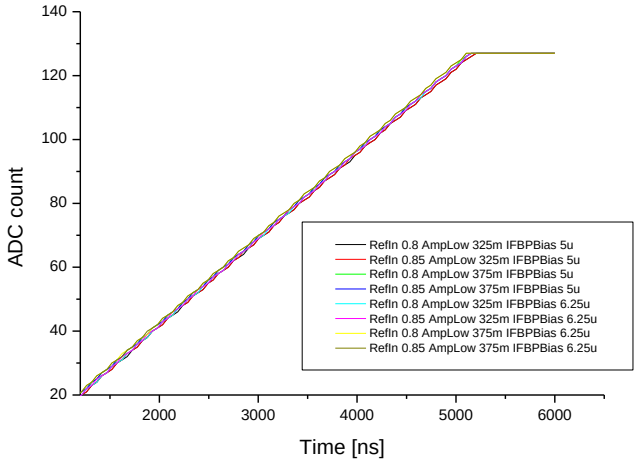

Figure 3: Transient simulation of ADC at doubled clock speed – different settings, typical transistor corner.

A signal current source with linearly- time-dependent current has been connected to the channel input. The current increases about 16uA/256 every ADC sampling interval. The ADC in simulation operates at 51.2ns sampling rate (normal DCD operation 100ns) – the clock is two times faster than in normal operation. Despite of this ADC works well in all the corners.

The simulation shows no effect of global voltage drops. This also agrees with measurements that do not show that ADC functionality depends on position in matrix. I would therefore skip the changes mentioned in review (page 11-16) (Reduction of RefIn current and introduction of additional bias DACs). These changes would make sense only if we had a hint that global voltage drops influence the operation, which we don't have according to presented simulations.

Missing codes

As mentioned in review the most probable cause of the missing codes is theoffset between the transconductors in the comparator. This offset is probably caused by a combination of the statistical (random) transistor mismatch and some systematic effect (e.g. transistor orientation, local voltage drop…).

First I will explain the simulations of the statistical transistor mismatch, and show that it can (in principle) lead to missing codes.

C++ simulation of the ADC algorithm where the noise and offset effects are added

Figure 1 shows one result of such a C++ simulation, which is just showing that an assumed comparator offset causes missing codes. A comparator offset of > 2uA (1/4 of the full positive range of one cell) is introduced in the first conversion step. The result is a missing code as sometimes seen in measurements.

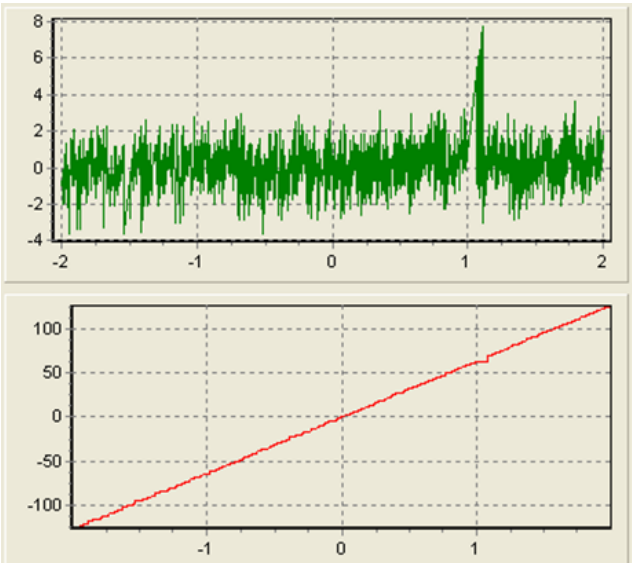

Figure 1: Missing code simulated in C++

The estimation of the offset between two transconductors starting from the values for the threshold voltage mismatch, from the UMC document "matching report".

Here I have derived a formula for the propagation of errors in the circuit containing two transconductors. The calculation has been done in C#.

The result is the following:

Mismatch (random offset) of the comparator current is 3.629uA (6 sigma value).

Notice: For a missing code, we need an offset of 2uA.

The question is whether this alone can cause missing codes.

With a 6 sigma value of 3.629uA (width of the distribution) an offset of 2uA happens quite seldom, however it is possible from time to time since we have a large number of ADCs (256) and every ADC has 8 x 2 comparators. (1024 comparators in total).

One important thing should be noticed:

In measurements (at least the measurements in Heidelberg) we mostly see the missing codes around +-64 and 0.

When we analyze the ADC algorithm we can see:

The mismatch in the comparator of the stage 1 produces missing codes around -64, 0, 64

The mismatch in comparator of stage 2 around -96, -16, 0, 16, 96, etc

We see mostly the missing codes -64, 0, 64, which means that mostly the first stage is affected.

If only random mismatch would cause the missing codes, then all comparators whould be equally affected and we would equally frequently see missing codes on many places.

There must be thus some systematic effect that makes the first stage worse in terms of offset. I will discuss possible effects later – some of them were mentioned in review (transistor orientation, local voltage drops).

Still, even if we have a systematic effect – we can improve the ADC by reducing the transistor mismatch. The systematic effect is obviously not the most dominant in total offset, otherwise all ADCs would have missing codes. Actually we see from the above calculation that the random transistor mismatch nearly can produce missing codes. It is probably so that a systematic effect makes about 10% of comparator offset and the random mismatch about 90%.

A few comments to the calculation of random mismatch

The design of the transconductor uses the triple-well TW-NMOS transistors, and two low-threshold voltage LV-PMOS transistors. The document "matching report" does not include the data for these

transistors, I have used for LVPMOS the data for normal PMOS and for TWNMOS the data for normal NMOS.

Further, in the “matching report” on page 45, it is written (the document is confidential, I can show it if you would like) that an NMOS had worse mismatch then specified (I used the specified numbers for the above calculation) if there was no protection diode through metal 1. Such a diode is missing at four important transistors in the transconductor. In the document,tThere is an example of 10um/1.5um NMOS that has sigma Vth mismatch of 1.02mV with and 2.79mV without the diode.

The above two facts (the missing models for TW- and LV transistors and the missing antenna diodes) could explain a larger mismatch and consequently more frequent missing codes.

I have further calculated which transistor mostly contributes to the mismatch. It is the NMOS current source NFB.

Notice, the above calculation yields to a larger offset than that I showed in review on page 24.The problem with the simulation was that I didn’t notice that the mismatch of LV PMOS transistors was simulated. I have redone the simulation and obtain very similar result as calculated above. Sigma mismatch is 575uA, or 6 sigma 3.45uA – Figure 2.

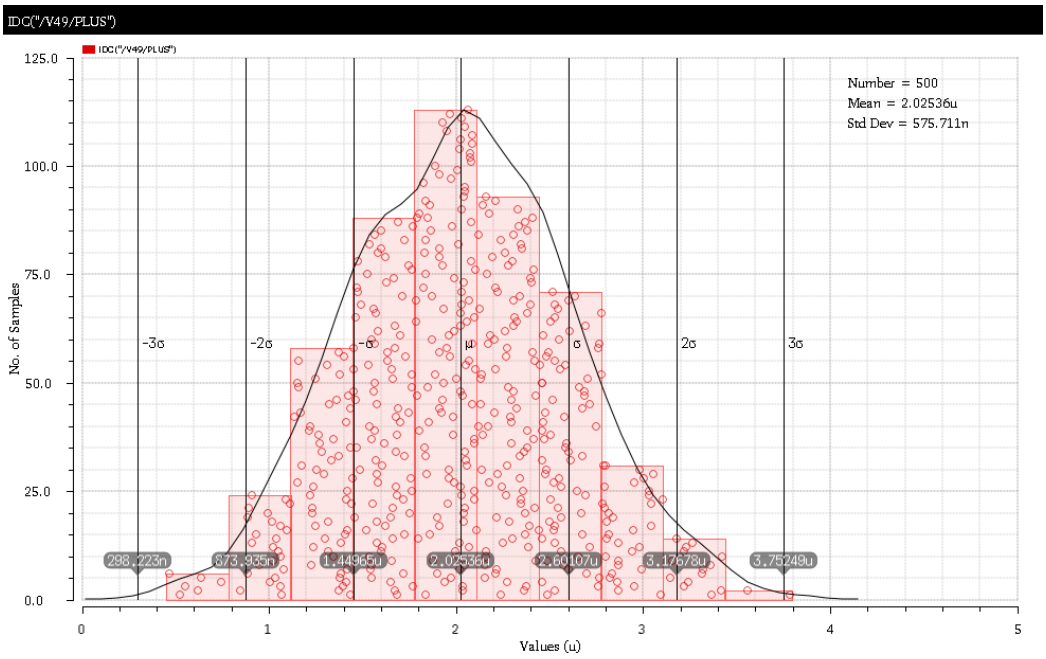

Figure 2: Repeated simulation of comparator offset.

I have done the Monte Carlo simulation of the full DCD channel (transient simulation). The simulation takes typically ~20 hours for 16 iterations.

As explained, when doing the Monte Carlo simulations I have the problem that TW- and LV-transistors do not have Monte Carlo models. I will try to obtain these models. Now, to cope with this, I have replaced the TW NMOS by normal NMOS and LVPMOS by normal PMOS with added ideal voltage source to mimic the different thresholds.

The result is the following, out of 16 ADC characteristics, one shows missing codes. This characteristic is shown in Figure 3 and 4. Only one code is missing at several places. To reduce the simulation time, only a part of the ADC characteristic was simulated.

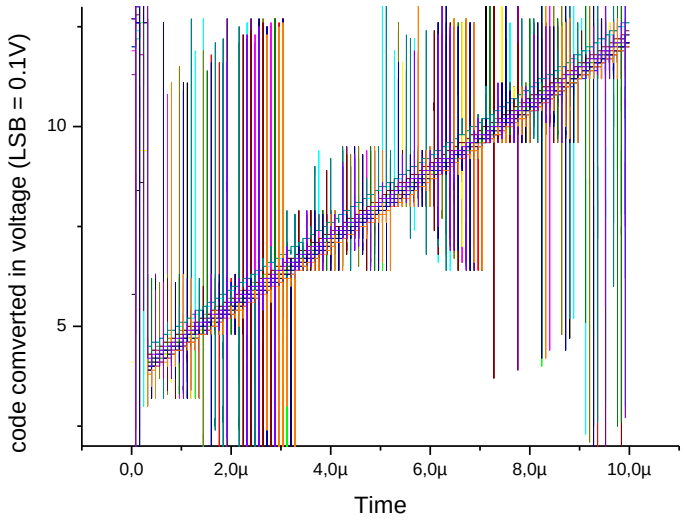

Figure 3: Monte Carlo Simulation of the DCD channel (all results), the spikes are due to conversion from digital code to analog voltage, when the code is not valid at CLK edge.

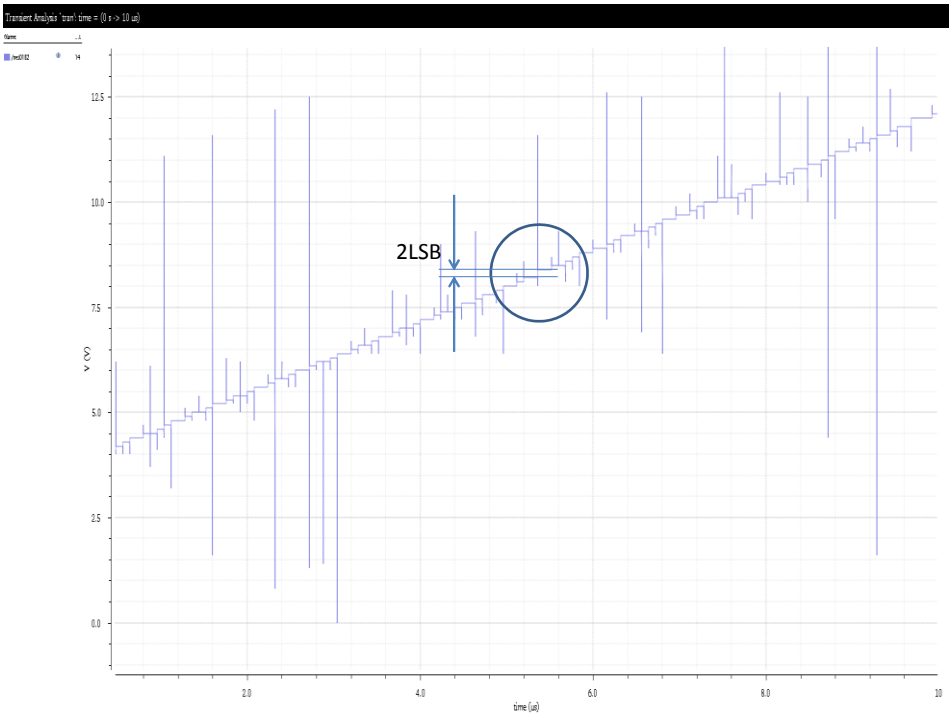

Figure 4: Monte Carlo Simulation of the DCD channel (result with missing codes).

In 20th B2GM I have proposed to increase the size of the NFB current source in order to improve matching and eliminate missing codes.

When I simulate ADC with the larger NFB source, no missing codes are observed.

However:

The fact that mostly the first stage is affected by the problem means that we have some systematic effect in the first stage.

There are several candidates:

Local voltage drop e.g. in RefIn or VDDA line can cause an offset that adds to mismatch.

Transistor orientation is not the same.

Missing dummy structures -one of the two transconductors that should match is placed at the edge of the layout. The environment for the edge-transistors is different (no dummy structures) which can add a systematic offset that adds to the mismatch.

I have estimated by calculation the local voltage drops and they are of the order of mV. This is not large enough to produce significant systematic offset (see e.g. simulation in page 27)

Transistor orientation and dummy structures are still good candidates.

One remark: Inserting of the large NFB current source will blow up the ADC layout so that there will be much less space for the decoupling capacitors in the channel. This could cause another type of problems (e.g. increased noise).

I am therefore proposing for the DCD design only to add the missing antenna diodes and the dummy structures.

The added antenna diodes will reduce the random mismatch and the dummy structures will reduce the systematic offset.

Notice, the missing codes are seldom when the DCD is configured properly. As example I mention my measurement from Heidelberg (hybrid 4 system), where on two chips (on each ½ of channels were measured) only one missing code with length > 3 was observed. Additionally, even a channel with a missing code is not useless. A problem occurs only if a DEPFET pedestal current is above the missing code, which is quite seldom and can be fixed by using 2-bit offset DAC. From this reasons I would not do complex layout changes that impose risk.

In the following text I address some changes and their necessity:

Parallel Sampling Mode

For the design I propose not to implement the parallel sampling mode. We are planning a chip submission within an engineering run. Tests with larger needle cards can be done more easily.

Protection Diodes for Digital IOs should be on VDDD

This is already implemented in the present chip.

Redo Protection Diodes for Analog Inputs

The present DCD as a 60 Ohm- poly resistor in series at the analog input. This resistor together with the present large protection diode and the VSource compliance should provide enough safety. Tests of VSource power supply, together with DCD are needed.

IPDAC range should be 60uA

The present IPDAC range is about 120uA – it was chosen so that it can emulate a pedestal current.

Notice 120uA is just the maximum range – the actual range can be adjusted 0-120uA in 127 steps.

The change of IPDAC from 120uA to 60uA range can be done easily.

Change the gain setting of the amplifier

The present DCD can operate with three gains settings. They correspond to the following signal ranges: 10uA, ~ 16uA, ~25uA

It would be desired that the DCD has the following range:

16µA, 22µA, 28µA, 35 µA

This can be done by using of feedback resistors 13.7k, 17.11k, 21.8k, 30k.

Adding of variable ranges is in principle ok, and will be done. However, it also introduces nonlinearity.

Therefore one possibility would be to choose the most optimal range after the long matrix measurements.

Notice, DPFET pedestal current can also be varied using Switcher Gate voltage and adjusted to DCD. Change in gq is often not large.

JTAG TCK change

The present DCD “samples” the data for the global and “pixel-” (channel-) register at the falling TCK edge and releases the data at rising edge. This was done deliberately on the first DCD to avoid setup and hold problems between the synthesized JTAG block and the full custom global- and pixel resistors. The scheme has been kept until DCD4pileine for compatibility reasons. DHP can cope with the sampling at falling edge since it has added possibility to invert TCK when writing into these registers. The scheme is working well on EMCM.

We will change sampling polarity for the pixel and global register to rising edge and the data output to falling edge. There is no need for a new synthesis of the digital block - the digital block has been already prepared for the last submission.

More complex test patters

For the design, I propose not to change the test patterns. We have problems in finding the optimal delays only because the sampling windows are very small. With improvements in DHP receiver, the sampling windows will be much larger and it will be much easier to find an optimal delay.

In the case of a second DCD design (can be submitted on an engineering run), where the digital block can be redesigned, more complex test patterns can be added.

Summary

List of changes

1. Digital transmission from DCD to DHP

I propose to remove entirely the hysteresis from the DHP receiver, to reduce the DHP input capacitance from 3-1pF.

The present DCD driver scheme is good.

The possibility to increase the bias current by switching additional bias resistor in parallel will be added to increase the distance between logic one and zero. (Comment IP – this change is not necessary, but it adds additional safety.)

2. Missing codes

Add four antenna diodes/transconductors and the dummy structures at the edge of the ADC

3. IPDAC range

The change of IPDAC from 120uA to 60uA range will be done

4. Gain settings

The present range settings 10uA, ~ 16uA, ~25uA can be changed to 16µA, 22µA, 28µA, 35 µA

(Comment IP – this change could be also done after the pilot run measurements)

5. Present DCD samples global control- and pixel-data at falling TCK edge; it will be changed to rising edge (Comment IP – this change is not necessary, but makes the design conform with the industry standard.)

The full verification procedure at the full chip level will be tried, when we have a stronger server available. Presently the QRC simulation of only one ´DCD channel takes ~10 hours.