Modern High-Speed Link Design

Jeremie David

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2017-69
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2017/EECS-2017-69.html

May 11, 2017

DAVID

Jeremie

Modern High-Speed Link Design

Capstone Project

Table des matières

I-Introduction

1-Project description

In the wake of the growing amount of data being processed in data centers, we want to be able to send those data faster and faster. The goal of this project is to build a modern High-Speed Link that can process data at a rate of 25Gb/s. Basically, we want to send data from a transmitter to a receiver. If we consider the straightforward solution of data transiting into a simple wire between those two blocks, we might encounter some problems. Indeed, at those speed, we have to cope with bandwidth limitation, noise, interference that will deteriorate the original signal. So, we aim at building specific tools at the receiver in order to recover this initial signal and get a clean received signal. Our infrastructure looks like the following one (Vladimir Stojanovic, 2004):

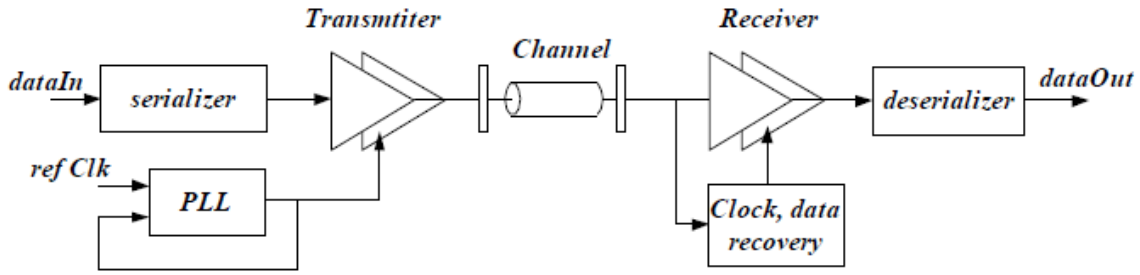


Figure 1.1: High-speed link block diagram

Let us analyze this structure. We want to send data from one chip. We will receive data coming from different part of the chip at the same time. However, all those data will travel through one channel. So, we need to serialize those incoming data packet. The serializer realizes this operation. The PLL will be responsible to generate the clock that set at which rate the data is sent into the channel. The data will be sent by the transmitter and will travel through the channel. The receiver will receive it. Inside this receiver, a specific architecture, described later in this paper, will try to recover the original signal that has been affected by the link. The data recovery block will try to synchronize its clock to the original one so that it can sample the signal when it is the cleanest possible. Again, we will go back on this point later in this paper. Finally, we will deserialize the signal and send it to the receiving chip.

So our goal is to provide a fully operational layout of the receiving block that compose this link with the constraint of operating at 25Gb/s.

2-Milestones associated to the project

In order to achieve our goals, we have divided our project into specific tasks, following a specific schedule. This part will describe those tasks and give a Work Breakdown Structure of our project.

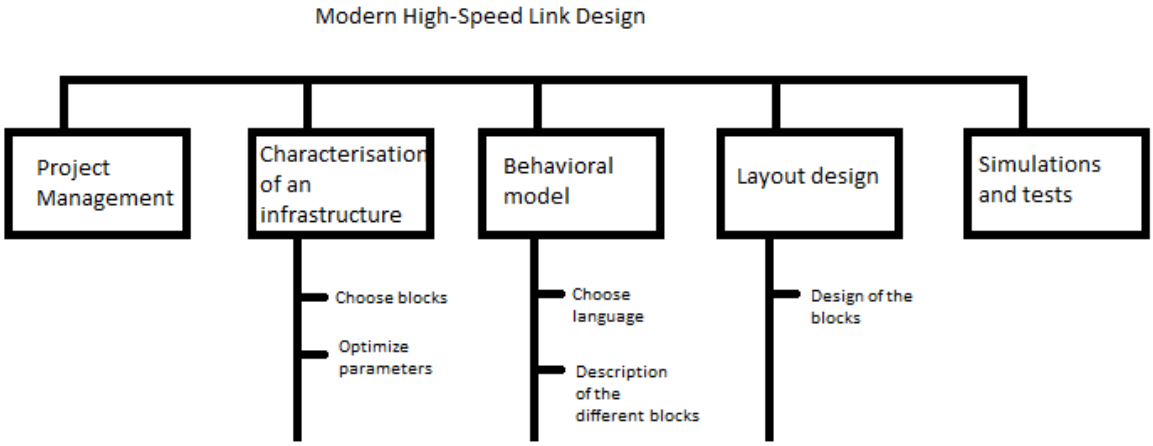
The first task was to go through different materials in order to gain insight about the scope of our project. We mainly focused on the thesis written by our advisor Vladimir Stojanovic. This part of the project enabled us to understand all the barrier that we will have to face (interference, noise…) in order to build a link that can operate at 25Gb/s. This was done in September/October.

The second part was to choose the different blocks we would need to build our link. To do so, we have a figure a merit called the eye diagram (see next section for more explanations) that can tell

us whether our received signal is clean or not. We used the Matlab code developed by the UC Berkeley team working on this project. This code provided us the different correction blocks at the receiver and the tools to assess the signal. Our goal was to optimize the parameters of the different correction blocks so that we can get a clean signal at 25Gb/s. Basically, we want to build the simplest link possible (with the less parameters possible) while accommodating our speed constraint.

The next part consists in designing the chosen infrastructure. My focus was on the design of the analog part which is made of the Continuous Time linear Equalizer.

The Final part is to design the layout. This corresponds to the physical view of the circuit on the chip. This task was achieved in April/May.



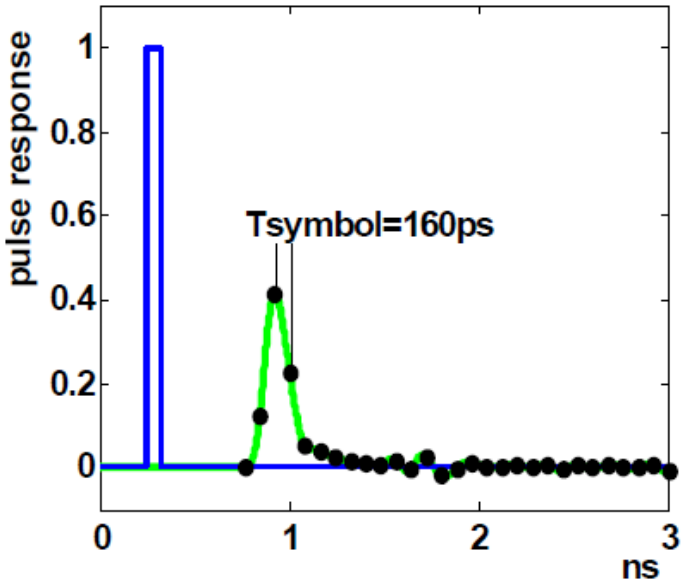
Work Breakdown Structure

II-High-Level behavior of the link

1-Causes of interferences

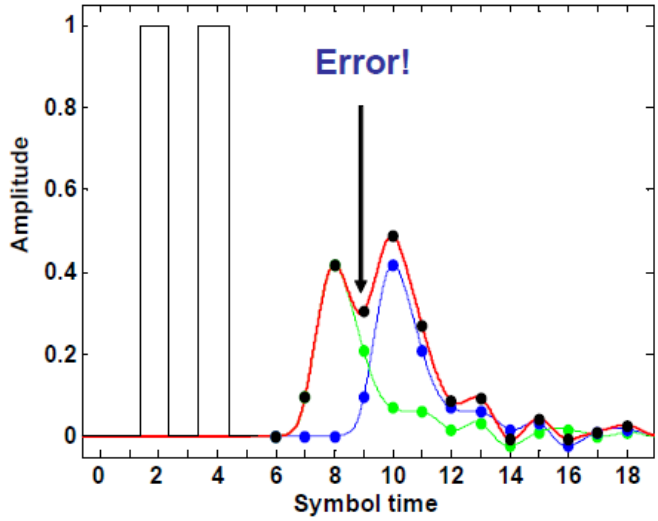
After reading some papers to gain insights about the project, we started working on the Matlab code developed by the UC Berkeley team. In order to get a better understanding of the issues the signal may face, let us consider the following example (Vladimir Stojanovic, 2004).

Let us assume that we want to send the following blue pulse signal. Due to bandwidth limitation, the received signal will be stretched and we will obtain the green signal. What we are really interested in is the 3rd dot, which carry the information. Other ones can lead to mistakes.



Pulse response

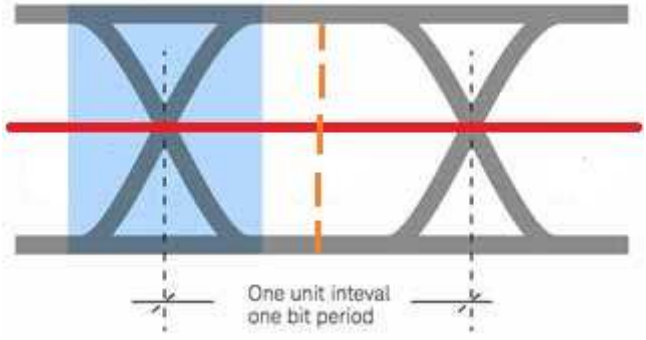
Indeed, let us consider the following case in which we send two pulses.



As we can see, the interference between the received signal will lead to a received sequence of bit 111 instead of 101. So our goal is to reduce the dispersion of the received signal in order to prevent those interferences from happening.

2-A way to assess the quality of our signal: the eye-diagram

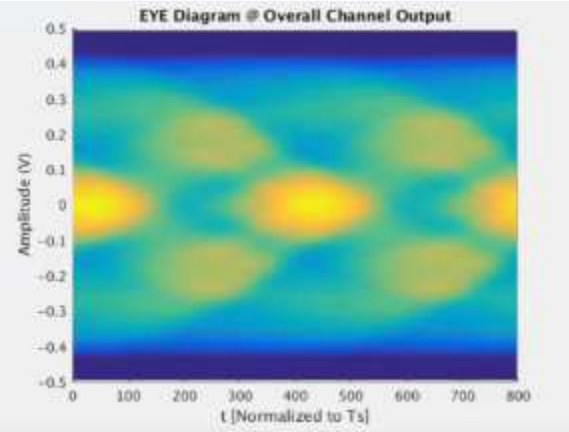
In order to mitigate this problem, we use three different infrastructures: CTLE, DFE and ZFE. Those blocks will be described later on. So our goal was to play around with the different parameters to get a neat signal using the least number of parameters possible. To assess our signal, we use the eye-diagram. Digital information is composed of 0 and 1 bits. Let us consider a sequence of 3 bits. The different combination possible are 000 001 010 011 100 101 110 111. If we superimpose the different corresponding transition, we obtain the following diagram:



Eye diagram

When we sample our signal, we want to be far from the transition so that we can determine if our bit is a 0 or a 1. Typically, we want to sample along the orange line. Indeed, at this time, if our signal is above the threshold voltage symbolized by the redline, our bit is a 1. Otherwise, it is a zero. This diagram is called an eye diagram due to its shape, similar to the one of an eye.

Nevertheless, this diagram is theoretical. For the speed we are working with, we encounter bandwidth limitation, interference, noise… This will imply our transition to suffer from jittering. Thus, without any correction (ie without CTLE, DFE, ZFE), our received signal looks like the following one:



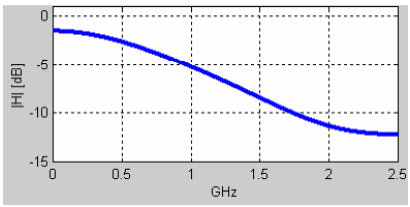
Received eye diagram

So the eye is closed. It is impossible to set a threshold voltage so that we can determine the value of our bit without any error. Our goal is to design the simplest correcting blocks in term of complexity that enable us to get a wide open eye diagram.

We optimized the different parameters of the CTLE, DFE and ZFE and amplifiers. We tested our link on a complex channel. I worked on the CTLE correcting block. Matthew worked on the DFE and Henry on the ZFE. Their paper give a detailed explanation of those blocks.

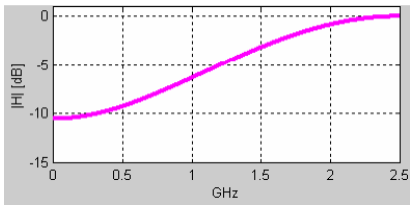
3-Principle of the Continuous Time Linear Equalization (CTLE)

The CTLE is a Continuous Time Linear Equalization. Basically, the channel is acting as a low pass filter. This means that the magnitude of the low frequencies component of a signal will stay the same but the high frequencies will be lessened. If we try to send a signal at a frequency of 25GHz, this signal will be hugely extenuated and it is impossible to detect it on the received side. The principle of the CTLE is to provide a high pass filter at the receiver in order to turn down low frequencies an amplify higher frequencies. In that way, we will be able to recover our original signal. This result can be summarized thanks to the following bode diagram (Vladimir Stojanovic, 2004):



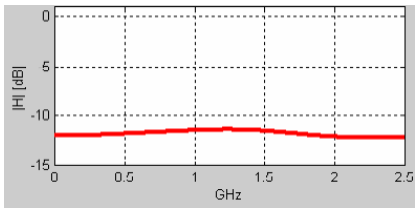
Low pass filter corresponding to the channel

+



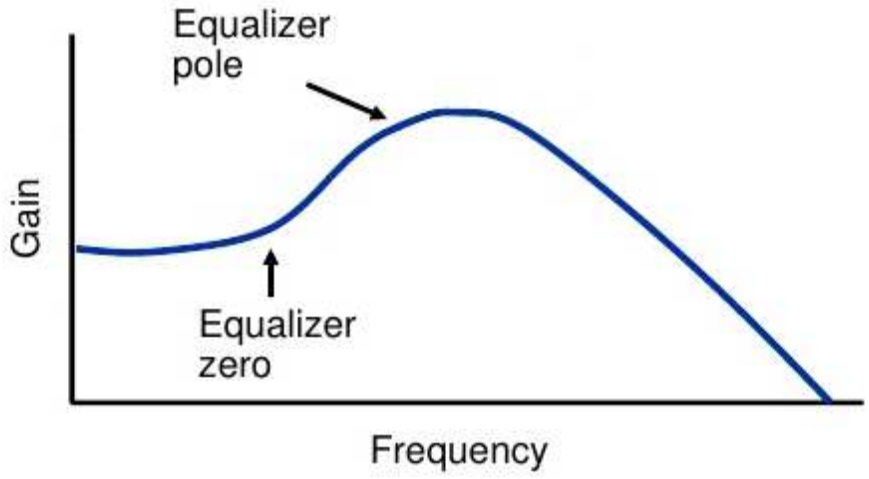
CTLE

=



Output signal

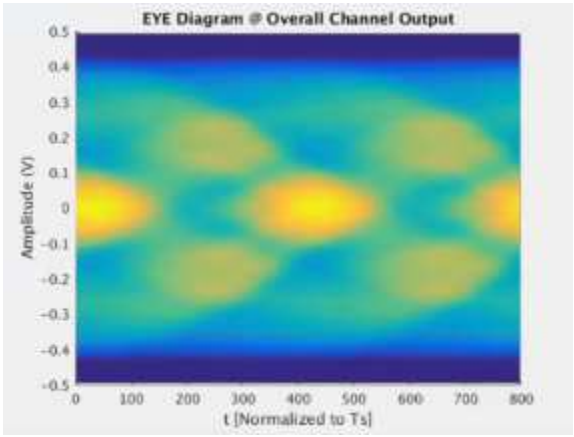
For this CTLE block, we can adjust three parameters. Let us take a look at the bode diagram of a CTLE.



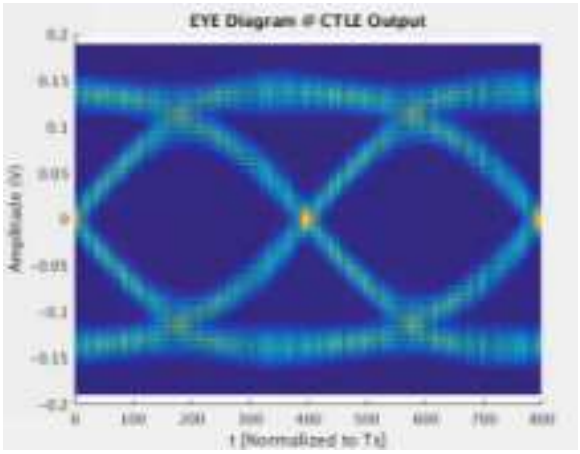
Transfer function of a CTLE

First, we can adjust the zero of our CTLE. This parameter will determine at which frequency the gain kicks in. Components above this frequency will be amplified. We can also set the poles that will be representative of the amplified frequencies. Finally, we can set the amplitude of our gain. This will determine the ratio of our amplified components over the low frequencies one. We have to be careful to not attenuate too much low frequencies. Indeed, if this happen, we will not be able to distinguish noise from the useful signal. We decided to use an amplitude of 2, to set the zero at 3GHz and to set the non-dominant pole at 17GHz.

You can see the effect of the CTLE on the following Eye diagrams:



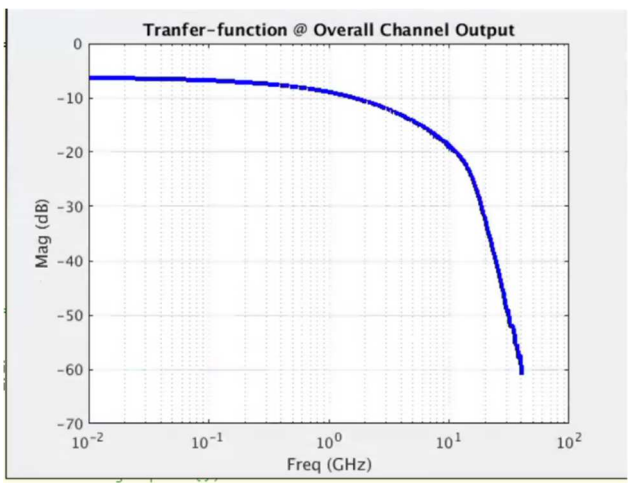
Without the CTLE

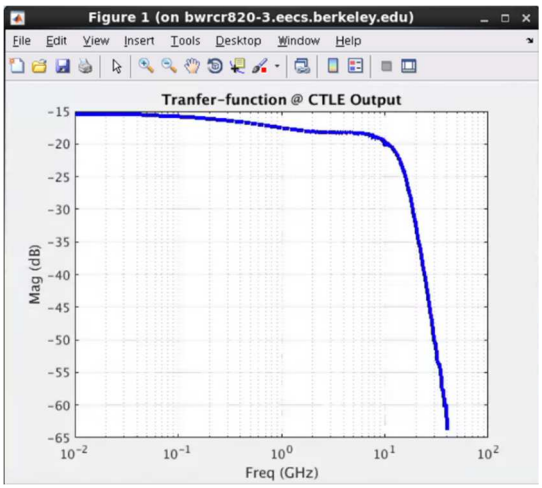


With the CTLE

Without the correcting blocks, our eye diagram is very fuzzy. You can see that the CTLE gives us a clean eye-diagram that enable us to detect the received signal.

We also compared the transfer function with and without the CTLE:

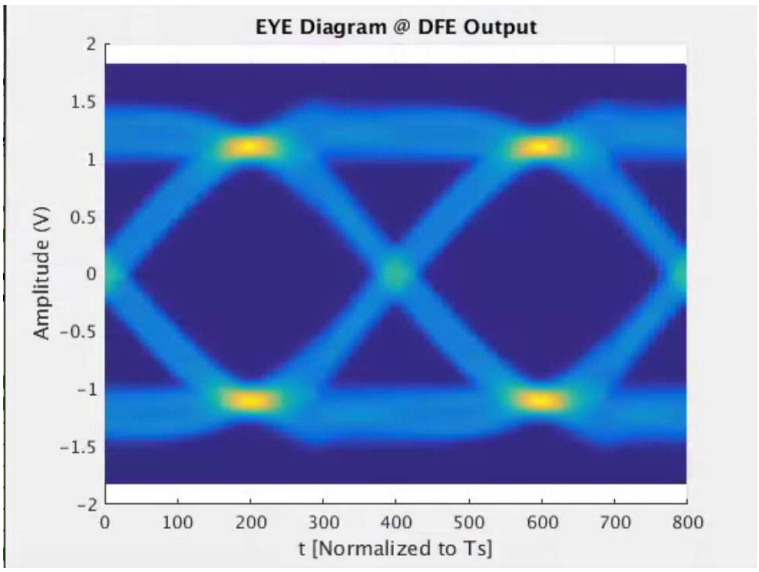




Transfer function without (left) and with (right) the CTLE

Without the CTLE, the gain starts decreasing at 1GHz. Adding the CTLE enables us to have a stable gain up to above 10GHZ.

Thanks to the DFE (see Matthew's paper) and the ZFE (see Henry's paper), our final eye diagram looks like the following:



Final eye diagram

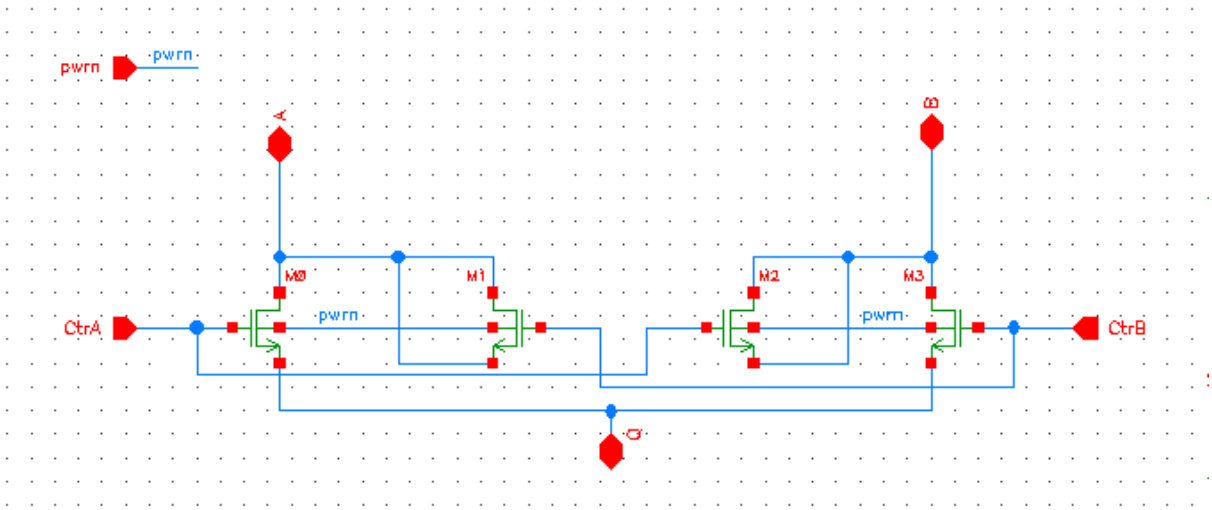
Thus, we have a very clean eye diagram and we will be able to clearly state if the received bit is a 0 or a 1 for a speed rate of 25Gb/s.

In order to achieve this task, one of our main problem was to set the limit of our design. Indeed, we could have designed very high end correction block that would give us very clean eye-diagram. But it will have a cost to develop this high performance infrastructure. So, our goal was to reduce the number of parameter we use while keeping a clean high diagram. We had to handle a qualitative problem. Indeed, we had to set a border that tells us when our eye diagram is clean enough. We had to brainstorm about the potential need of each correcting block. After running different simulation, we decided to use the three correcting blocks (DFE, CTLE, ZFE) and one Op Amp.

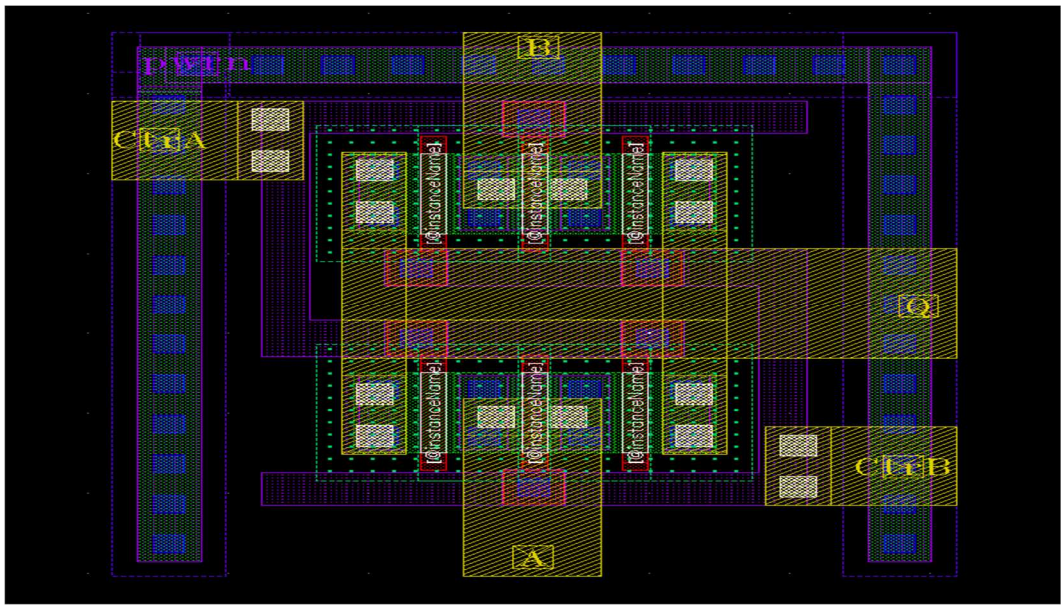
Our next step was to focus on the design of the different module. At this point, our team split. Indeed, if so far we worked together in order to optimize our link, now we will work on different tasks. Each one of us worked on the design of a specific block. We used the Matlab models we settled for the different part of the circuit and translate the characteristics chosen into an actual design.

III - CTLE Design

This part is will focus on the actual design and implementation of the CTLE. Basically, when we want to build a new circuit, there are four major steps. All of them are achieved with the software Cadence Virtuoso, which is the most common one in the field of analog design. The first step is to draw the schematic of the circuit. Then, we run different simulation to ensure that the circuit meets the specifications in term of bandwidth and signal magnitude. The third part consist in designing the layout of the circuit. This means that we are going to translate our circuit into its actual disposal. This is how the different elements will be printed on the silicon wafer. This layout is what will be sent to the foundry. The diagram below might help to understand the difference between schematic and layout.



Schematic of a basic circuit with 4 transistors



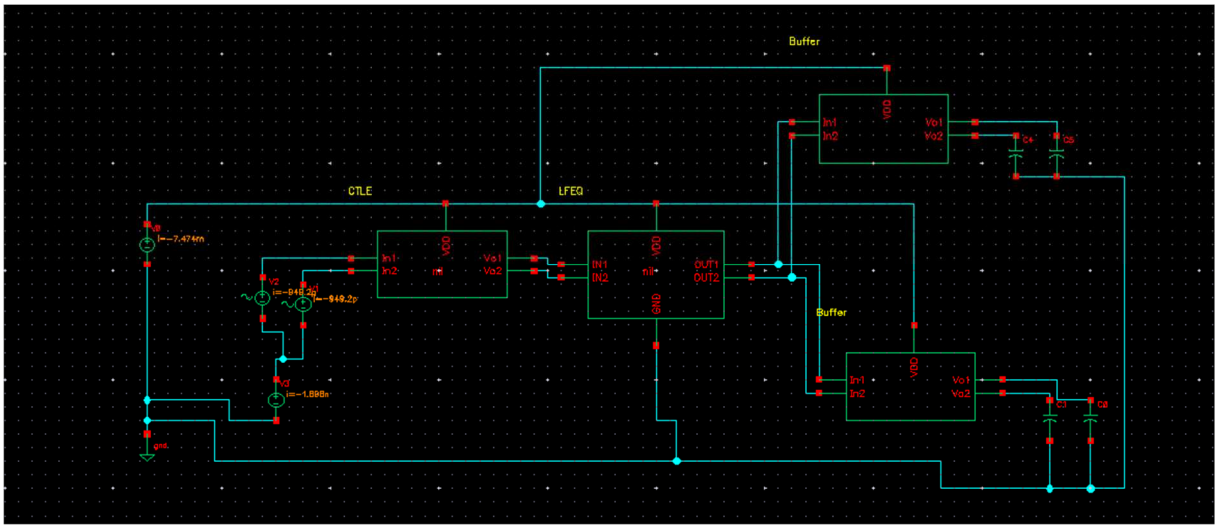
Layout of this circuit. The transistors are in the green part. The yellow and purple areas are the metal connection between the different part. This layout corresponds to how a circuit is printed on a silicon wafer

The last task is to run the post layout simulations. In this part, the software includes the parasitic elements due to the layout. Indeed, in the schematic model, all the wires are ideal for instance. However, in the layout, we draw each specific metal wire to links two components. So now, the software can have access to the characteristic of this wire (length, width…) and can assess its parasitic characteristics (resistance, capacitance…). Therefore, after running those simulations, we need to ensure that the circuit still meet the specification.

1-Schematic

a-Overview of the design process

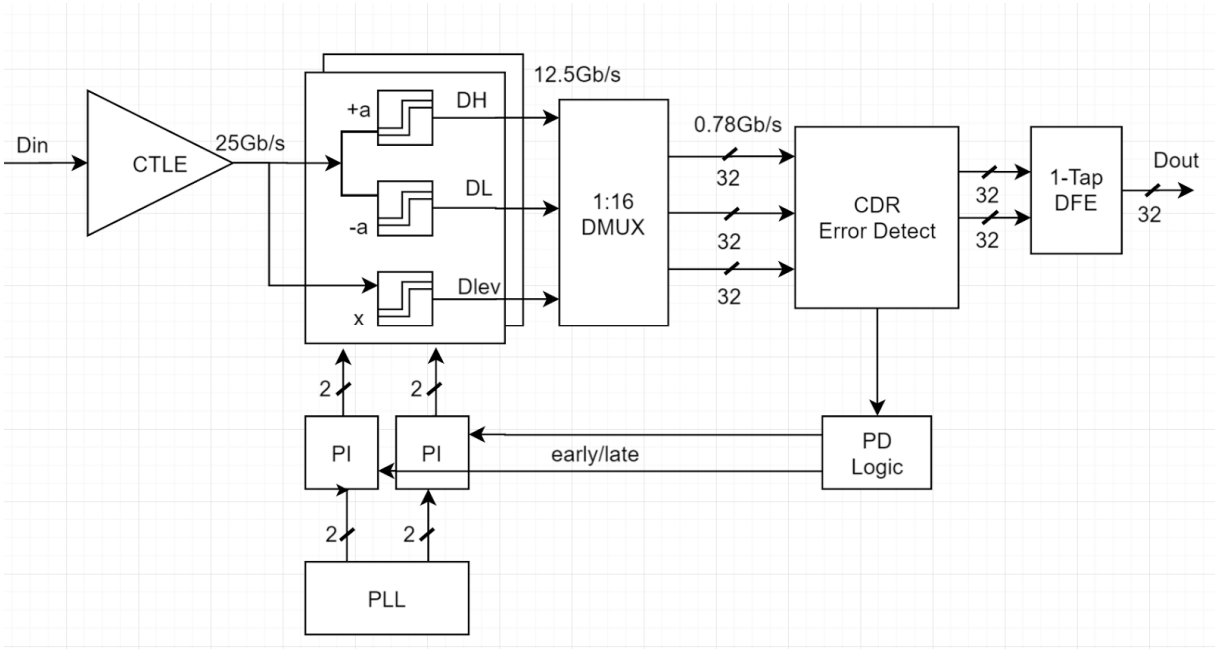
The whole CTLE is made of five blocks. The corresponding circuit is the following one:



CTLE

The name of the blocks appear in yellow. There are a CTLE, a LFEQ, and 2 buffers. As stated in the previous part, the CTLE will amplify the high frequency signal, for frequencies between 3GHz and 25GHz. Besides, we also saw in part II that we faced small losses (order of magnitude of 3.5dB) for frequencies above 500MHz. In order to mitigate this effect we add a low frequency equalizer (named LFEQ on the previous schematic). Basically, this equalizer will act like a CTLE. It will give a lower gain but will amplify a larger span of frequencies.

One last important point is the presence of the buffers. The CTLE will drive the rest of the circuit. But it actually 'sees' the input of the next module which is the strong arm (cf Matthew's paper). Indeed, the overall infrastructure of our project is the following one:



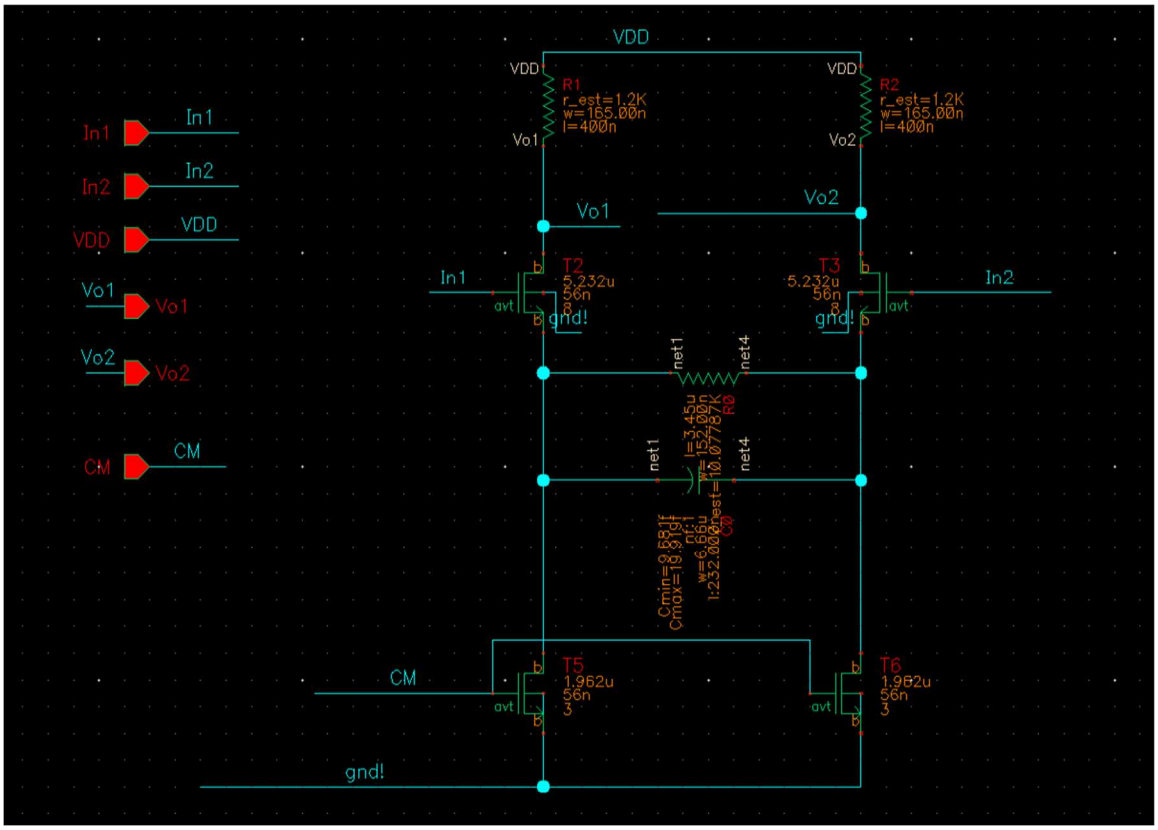
Infrastructure at the receiver

So, the CTLE sees two blocks of three comparators (+a,-a and x). The input of each block is equivalent to a capacitance of 15fF. The total capacitance is 2*15=30fF. However, when we design this type a circuit, we need to find a compromise between high gain and high bandwidth. Those capacitances of 15fF are high and they will define the bandwidth of our circuit. If we use just the CTLE and the LFEQ, we will not be able to have a 25GHz bandwidth without attenuating the signal by a lot. Therefore, the principle of the buffer is to 'insulate' the CTLE/LFEQ from the rest of the circuit. In this case, the buffers will lessen every frequencies (up to 25GHz) by 4dB. At a first glance, those buffers seems to be useless because the just extenuate all the frequencies. However, now the CTLE and the LFEQ 'see' the input of the buffers. Those inputs are made of 5fF capacitances. This is 3 times lower than the initial 15fF. So now, we will be able to get a high gain and the desired bandwidth.

In the design flow, we start by designing the last bloc (Buffer) and then the previous ones (LFEQ then CTLE). Indeed, we now the loading capacitance, which is the 15fF capacitance that we want to drive. So we can design the buffer. This will give us a size for the input capacitance of the buffer and we will be able to design the LFEQ thanks to this value. However, the CTLE being the main component of our circuit, we will start by introducing this module.

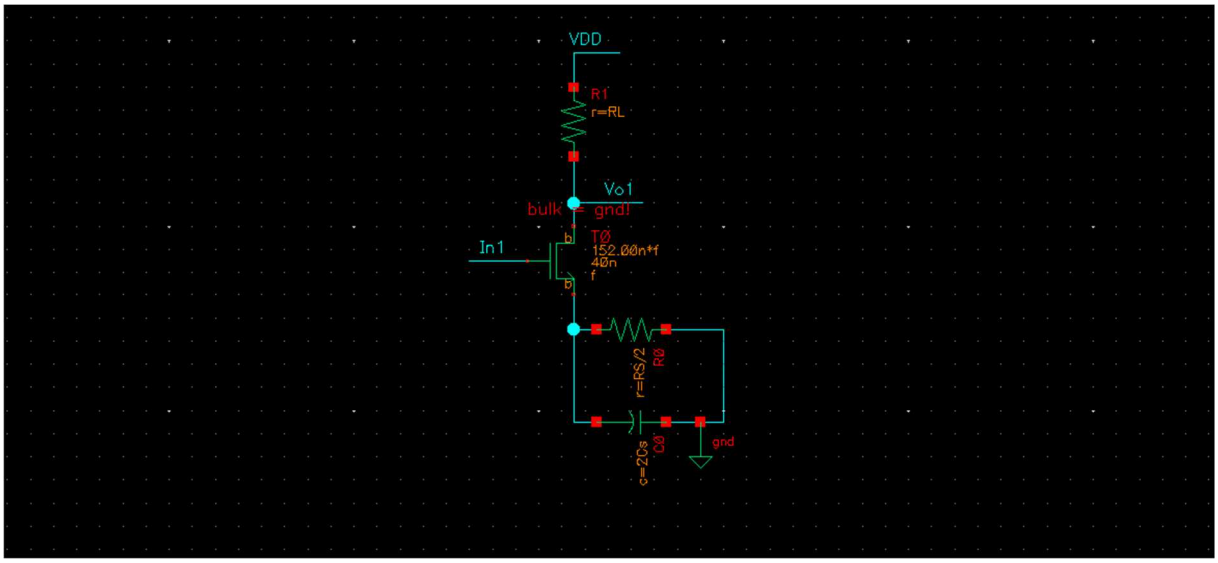
b- Continuous Time Linear Equalization

As explained in the previous part, the CTLE will be characterized by its zero – the frequency at which the magnitude increases- its non-dominant pole – the frequency at which the gain decreases – and its peaking gain. The corresponding circuit is the following one (Parikh, 2013):



CTLE

In order to analyze this circuit, let us draw the equivalent half-circuit for the differential mode.



Half Circuit for the differential mode

For this circuit, the transconductance is

$$Gm = \frac{gm}{1 + gm.(\frac{Rs}{2} \;//\; \frac{1}{2sCs})}$$

Where gm is the transconductance of the transistor.

Thus:

$$Gm = \frac{2.gm.(1 + sRsCs)}{2 + gmRs + 2s\,Rs\,Cs}$$

The output resistance is:

$$Rout = Rl \;//\; Cl$$

And:

$$Rout = \frac{Rl}{1 + s\,Cl\,Rl}$$

So, the gain of this circuit is:

$$av = Gm\,.Rout = \frac{2gm\,Rl\,(1 + s\,Rs\,Cs)}{(1 + s\,Cl\,Rl)(2 + gm\,Rs + 2s\,Rs\,Cs)}$$

Thanks, to this expression, we can derive the DC gain and the location of the poles and zero.

The DC gain is:

$$Adc = \frac{gm\,Rl}{1 + gm\frac{Rs}{2}}$$

The zero is:

$$f_0 = \frac{1}{2\pi\,Rs\,Cs}$$

The first pole is:

$$p_1 = \frac{1 + gm\frac{Rs}{2}}{2\pi\,Rs\,Cs}$$

The second pole is:

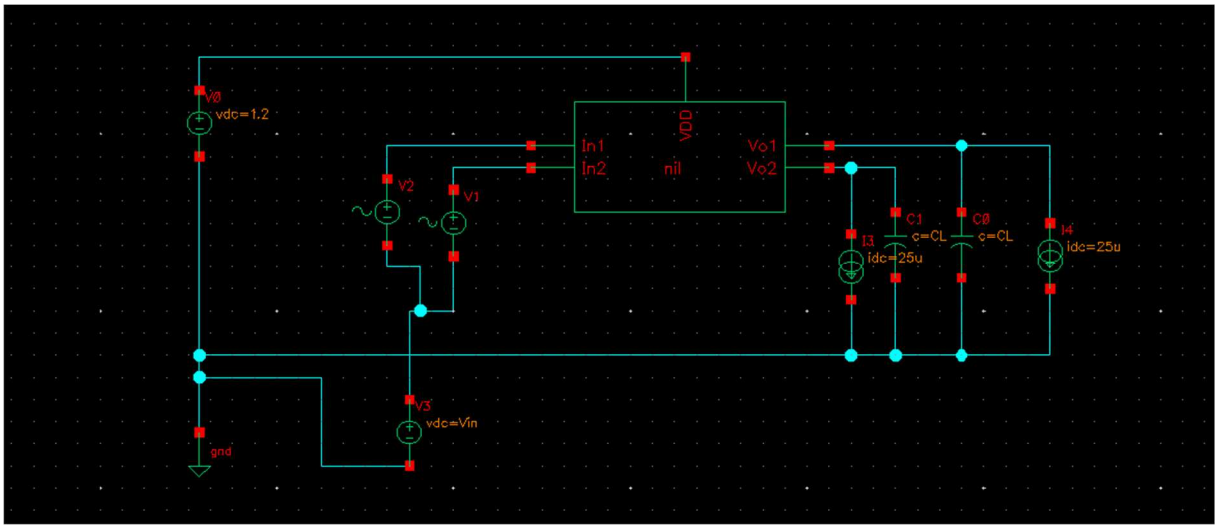
$$p_2 = \frac{1}{2\pi\,Rl\,Cl}$$

The peaking gain is:

$$a_{peak} = gm\,Rl$$

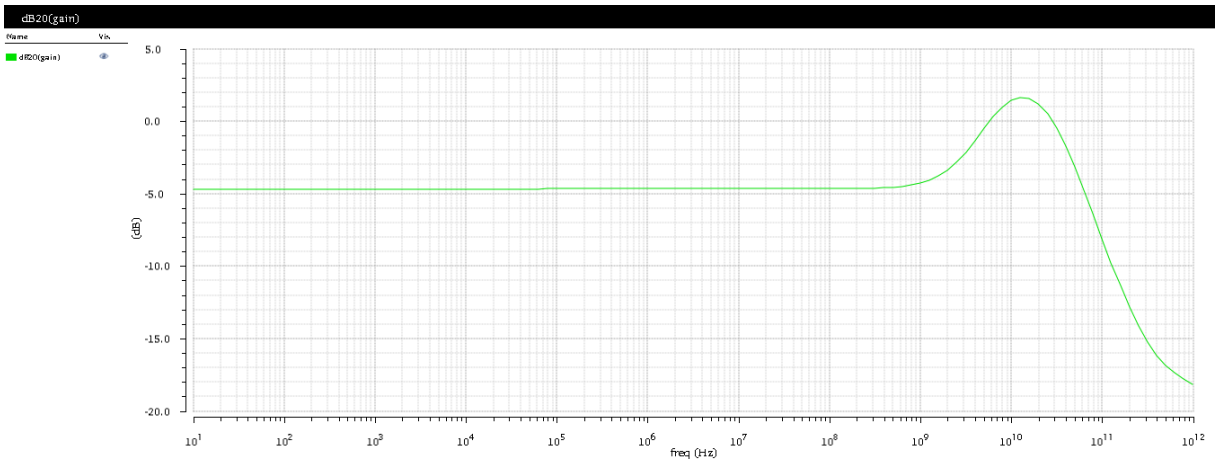
In the last part, we established that our zero would be at 3GHz, our second pole would be at 17GHz and our peaking gain would be 2. The loading capacitance (provided by the LFEQ) is Cl=4.3fF. Thanks to this, we can set our different parameters (Rs, Cs, Rl and gm).

We then simulated this circuit to ensure that it meets the specification. The Testbench circuit is the following one:



TestBench for the CTLE. The central bloc is the CTLE. The capacitances and the current sources on the right represent the following stage (LFEQ).

As a result, we obtained this Bode Diagram:

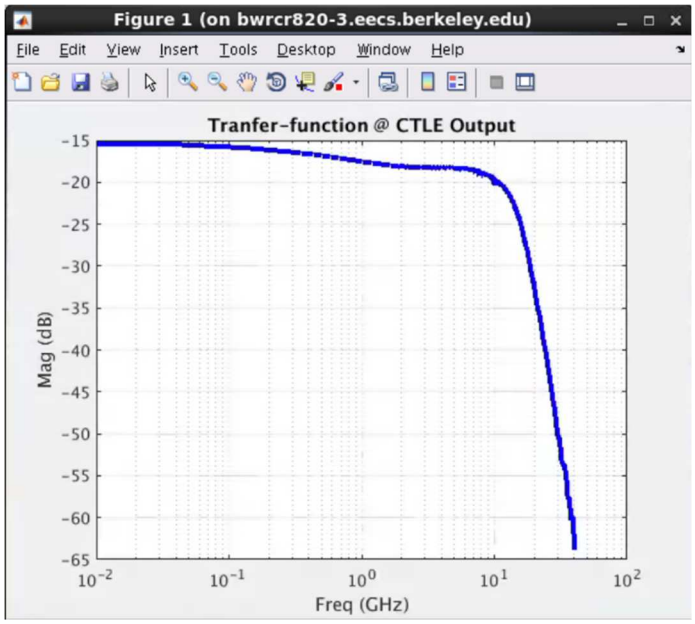


Bode diagram for the CTLE

As we can see, our gain starts increasing at 3GHz and decreases around 17GHZ. Besides, we have a 6.3dB peaking gain, which correspond to a magnitude of 2 (20log(2)=6dB). Thus, so far, our circuit meets the specification. Let us now consider the following block: the low frequency equalizer.

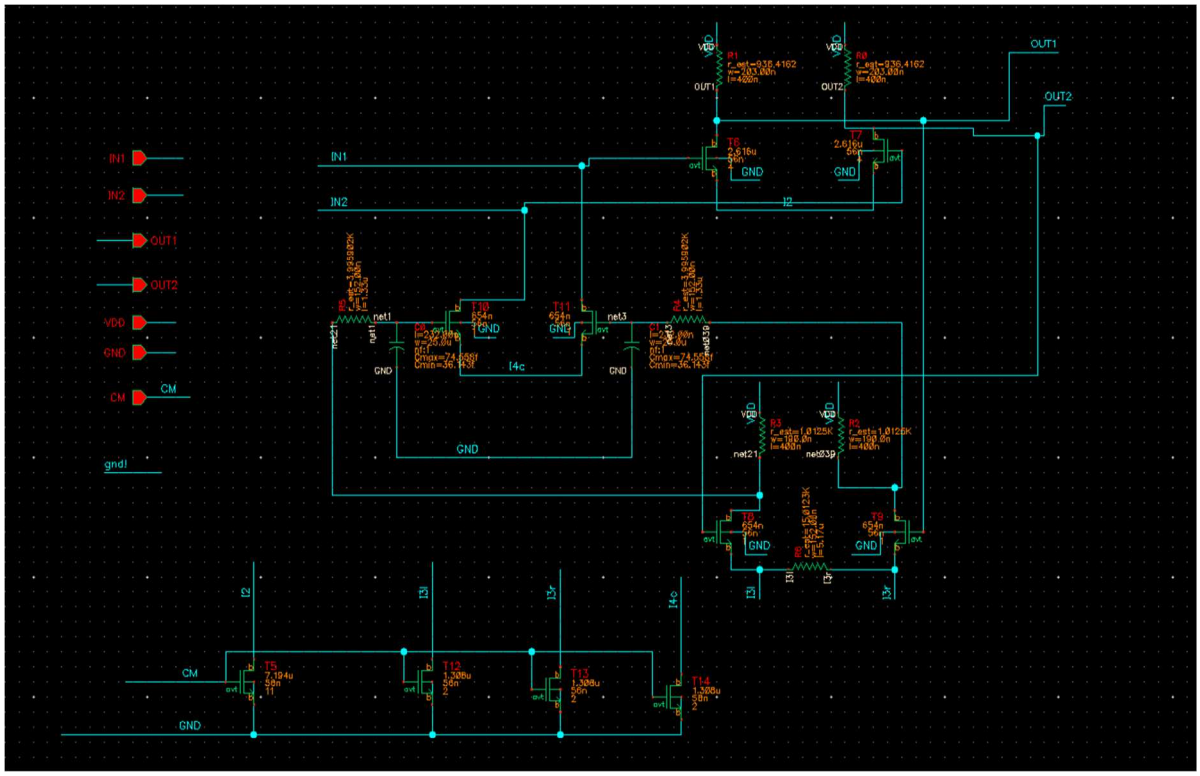
c-Low Frequency equalizer

To understand the role of the LFEQ, we need to go back to the bode diagram of the previous part. This is the transfer function of the signal corrected by the CTLE:



Transfer function of the corrected signal

As we can see, even after correction, we still have losses of 3.5dB for low frequencies (from 600MHz to 20GHz). The principle of the low frequency equalization is to increase the gain by 3.5dB over this large band. The circuit is the following one (Parikh, 2013):



Low frequency equalization

The analysis of this circuit is a little bit more tedious than the previous one. The derivation is introduced in the annex. The closed loop gain of this circuit is:

$$A_{cl} = \frac{-gm_2.Rl_1.Rl_2.(1+\frac{gm_3Rs_3}{2})(1+sC_{fb}.\left(Rl_3+R_{fb}\right))}{gm_4.Rl_1.gm_3.Rl_3.gm_2.Rl_2+(1+\frac{gm_3.Rs_3}{2})(1+sC_{fb}.\left(Rl_3+R_{fb}\right))(1+s\ Cl\ Rl_2)}$$

So, the DC gain is:

$$A_{dc} = \frac{gm_2.Rl_1.Rl_2.(1+\frac{gm_3Rs_3}{2})}{gm_4.Rl_1.gm_3.Rl_3.gm_2.Rl_2+(1+\frac{gm_3.Rs_3}{2})}$$

Our zero is:

$$f_0 = \frac{1}{2\pi\ C_{fb}.\left(Rl_3+R_{fb}\right)}$$

If we make the approximation $Cl\ Rl_2 \ll\ C_{fb}.\left(Rl_3+R_{fb}\right)$, then, the first pole is:

$$p_1 = \frac{1}{2\pi\ C_{fb}.\left(Rl_3+R_{fb}\right)}(1+\frac{gm_4.Rl_1.gm_3.Rl_3.gm_2.Rl_2}{1+\frac{gm_3.Rs_3}{2}})$$

The second pole is at:

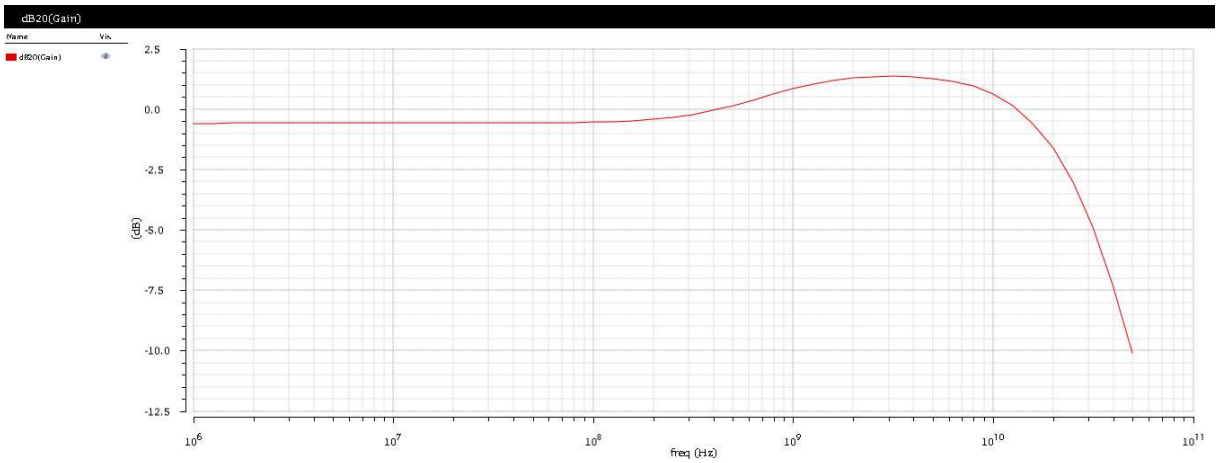
$$p_2 = \frac{1}{2\pi\ Rl_2\ Cl}$$

The peaking gain is:

$$a_{peak} = gm_2Rl_2$$

For this circuit, we want our zero to be at 600MHz, our second pole at 17GHz and a peaking gain of 3.5dB. The loading capacitance (provided by the two buffers) is 10fF.

We sized our circuit thanks to those parameters. The results of the simulation for the LFEQ are introduced below.



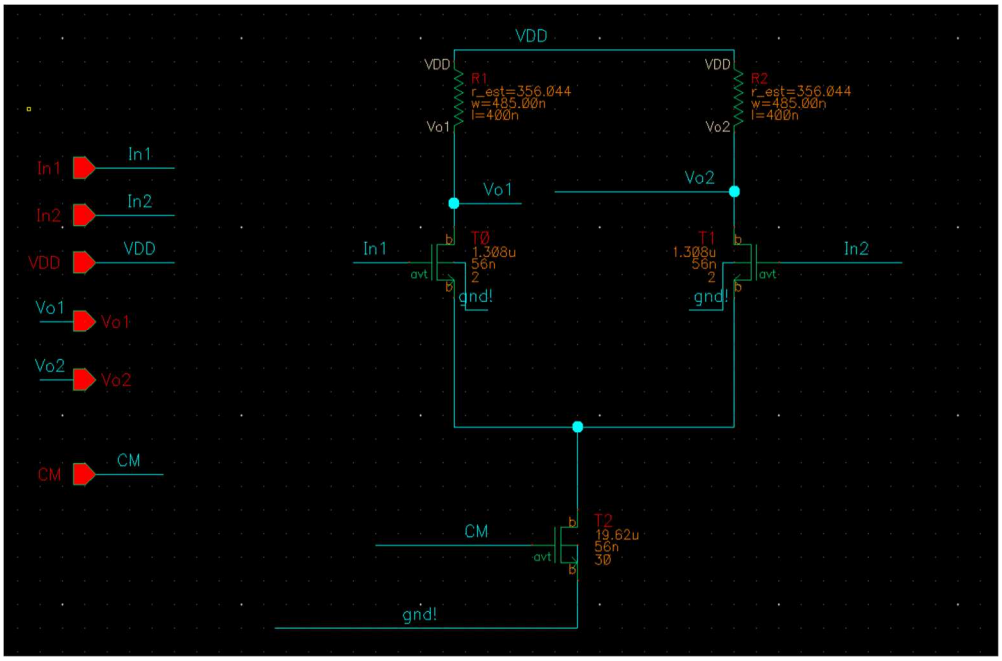
Transfer function for the LFEQ

Our peaking gain is 3.5dB, the zero is at 600MHz and the non-dominant pole is at 17GHz. So the circuit meet the specifications.

d-Buffers

Let us now consider the buffers. As explained in the introduction, those buffers will enable us to decrease the capacitance that the LFEQ has to drive. Another important point is that the signal at the input of the CTLE has a 500mV magnitude. For the next stage, which is the strong arm (cf Matthew paper), we want a 125mV amplitude. Thus, there is a factor 4 of attenuation, which corresponds to 20log(4)=12dB. So, our overall circuit must have a DC gain of at least -12dB. We already know that the DC gain of the CTL is -4.7B and the gain of the LFEQ is -0.5dB. So, the gain of the buffers need to be greater than -6.8dB.

The circuit corresponding to the buffers is straightforward.



Circuit for the buffers

The gain for this circuit is:

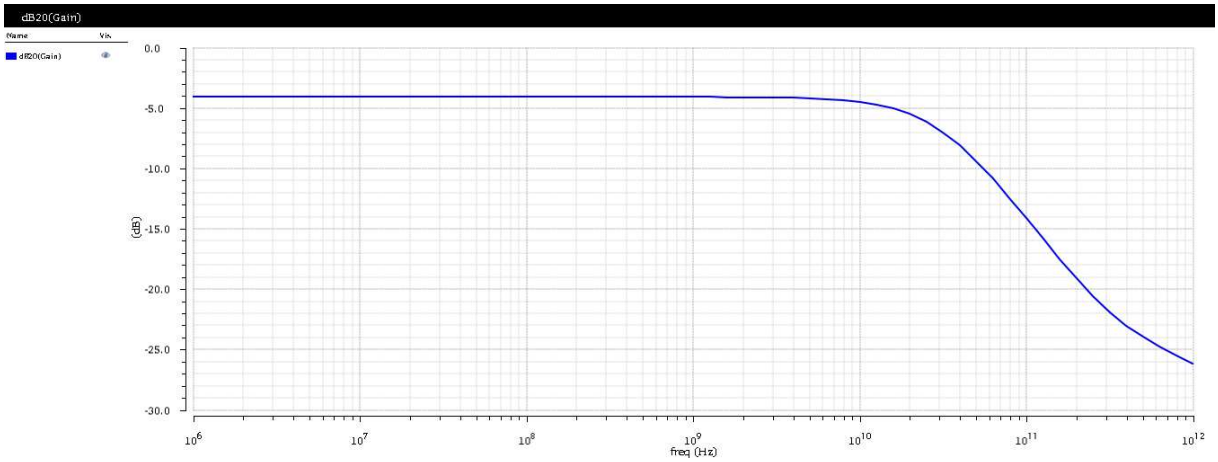
$$av = Gm\,.Rout = \frac{gm\,Rl}{1 + s\,Rl\,Cl}$$

We just have one pole at:

$$p_1 = \frac{1}{2\pi\,Rl\,Cl}$$

We want this pole to be as far as possible. We set it at 30GHz so that it does not affet the gain of the previous stages. The loading capacitance (provided by the strong arm) is 15fF. We sized this circuits to reach these specifications.

The transfer function is:

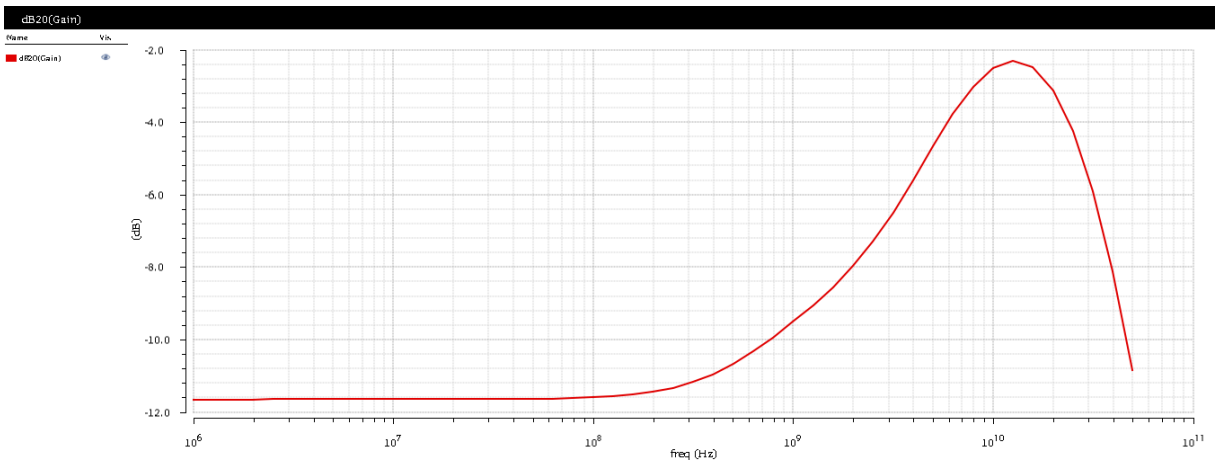


Transfer function of the buffer

The pole is at 30GH and the DC gain is -4.1dB.

e-Overall gain

Now that we designed the different blocs, we can simulate the behavior of the overall equalizer. The transfer function for this whole bloc is:

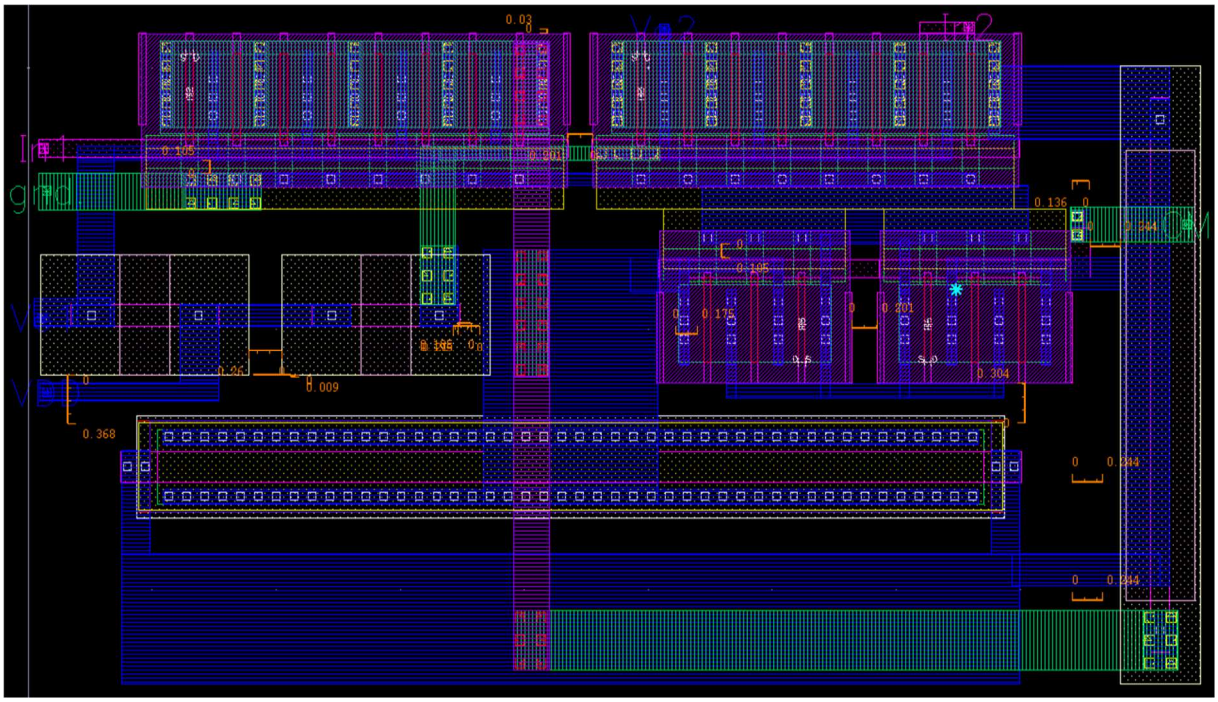


Total transfer function

The DC gain is -11.5dB, which is greater than the -12dB required. The total peaking gain is 9.3dB. We have a zero at 600MhZ, another one at 3GHz and a double pole at 17GHz. Eventually, this bloc meet all the specifications we set.

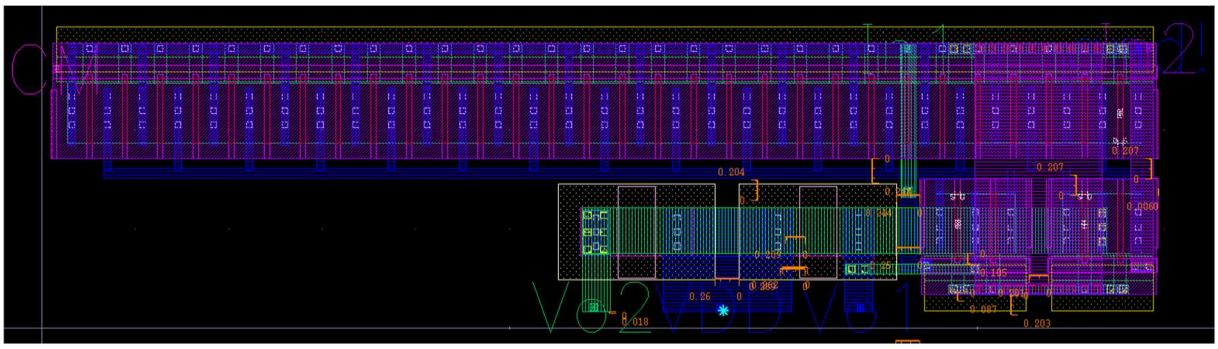
2-Layout

The next step of the design flow was to do the layout corresponding to the actual circuit. This is how the circuit will be printed on the silicon wafer. When designing this circuit, we have to ensure than the circuit has no DRC errors. This means that the circuit does not violate any rule related to the printing. Besides, we have to verify that it passes the LVS (Layout versus Schematic). This test check that the layout corresponds to the schematic we designed before. All the layout introduced passed the DRC and LVS tests.

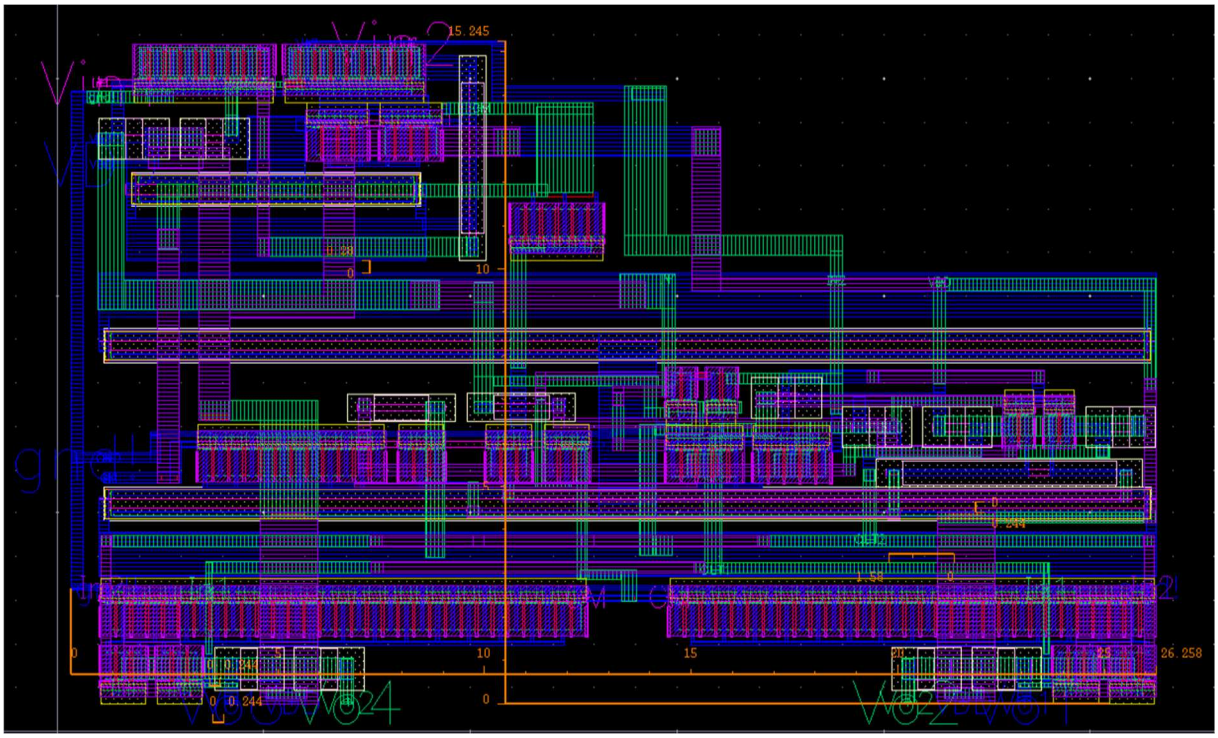


Layout for the CTLE

Layout for the low frequency equalizer



Layout for the buffers



Layout for the overall CTLE

The size of the final design is 15.245µm x 26.258µm.

IV - Engineering Leadership

1-Introduction

In this part, we will discuss how we structured the management roles of our group as well as the division of labor for the technical hardware blocks in the link. Besides, we will highlight the fact that when we build the link, we need to be aware of the IP surrounding the technologies we are building with and the legal contracts we are bound to. Finally, we will discuss the applications of our work within the University and future projects this one makes possible.

2-Project management

The different blocks that we are building cover different fields. Indeed, the signal received is analog. It will then be processed and converted into a digital one. Then, other processing techniques will be applied on this signal. Thus, this kind of project requires a broad knowledge and skills in the signal-processing field. Three main areas stand out: analog design, mixed signal design and digital design. This kind of project perfectly fitted our complementary knowledge as each one of us is specialized in one of this field. After a general overview and design of the global infrastructure, our team split to dive into each one of those parts.

3-Industry Analysis

We are not going to launch our product on the market. Indeed, our project is more focused toward the academic field. However, if we want to understand what drives it, we need to take a look at the current market.

As we stated before, there is a growing need of high-speed data links. Currently, this project can subsist the needs of two main field of interest. The first one is the classic modern chip design. As we increase the rates at which data transit inside a chip, we need to provide links that can send data from one chip to another. This link has to endure those speeds without deteriorating the quality of the initial signal. The main competitors on this branch are the big tech companies like Intel, Google, and Apple.

The second field of application are the data centers. During the last decade, we have seen the expansion of the flourishing industry of cloud computing. Indeed, we can see that TE Connectivity, one of the biggest provider for backplanes, which are the channel between the linecards inside a server, has an healthy situation, generation revenues of $2.4 billion in 2016 (Darryle Ulama, 2016, p26). Those data centers are composed of racks on which chips are linked through backplanes. In the wake of the general trend described above, backplane links has to face an increasing demand of data speed. The IBIS report on search engines in the US specifically highlights this point (Taylor Palmer, 2016, p34). In order to remains efficient and up to date, search engines has to constantly index every web pages. Due to the huge amount of data, a large bandwidth is require in order to offer instant search to everyone. The equipment deployed to cope with this are more and more complex and the current capacity of Google's data centers is assessed at 1.13 petabits/s, which is 100 times more than ten years ago. Besides, during the last 6 years, the bandwidth has been multiply by a factor 50. This increasing bandwidth demand leads us to develop links that are able to fill this demand so that the industry can keep growing in the following years.

Despite this important demand, we cannot manufacture and sell our product. The costs of manufacturing in this high-end industry are very high and are prohibitive for small companies. That is why we try to focus on a specific side of the development process in order to reduce the cost of it. We will just design the architecture of this link. This will enable us to fill the needs of the academic branch.

4-IP issues

Unlike small social media startups, the way we try to commercialize our products is very different and finally the buyers we target may not to be the entire public. Since we are not building the real "hardware" infrastructure, which both needs a lot of infrastructure and is relatively impossible for a small entry startup. We decide to take a similar approach as ARM.inc does in the semiconductor industry. As we all know, ARM is one of the few chip companies that actually never manufacture chips, instead they just design the architecture IP and sell the IP to other semiconductors giants such as Intel and Apple. Similarly, we try to generate our own high-speed link IP and sell it to our target tech companies who need to research on their own data-center/chip infrastructures. In this way, we don't have to confront with the huge cost comes with manufacturing our new high-speed design, which is both impossible for a startup and easily retaliated by the industry leaders like Intel as soon as our product is released since it simply has more resources. Moreover, our design mainly stems from Berkeley research groups and the group of people who know and have experience in our cutting-edge technology is relatively small, the chance of being substituted is relatively small given the unique nature of our product.

5-Academic

Nevertheless, the initial purpose of our project is not to launch a product on the market and compete with the other stakeholders. Indeed, due to the high constrains, it takes a long time to design a link that can operates at those speed. Big companies are already working on it and the price of their product can reach several tens of thousands of dollars. However, some engineer need those links to develop their product. Thus, our main goal is to provide an accessible technology to them so that they can work with it without spending time on developing their own link. In that way, the library we will deliver at the end of our project more echoes the needs of the academic field than the one of the industry. The main advantage of this strategy is to provide at cheap library of high speed links.

In that way, other researches will be able to rely on this infrastructure. For instance, if they want to assess the quality of the signal they would have to cope with after receiving the signal, they can use our library to do so.

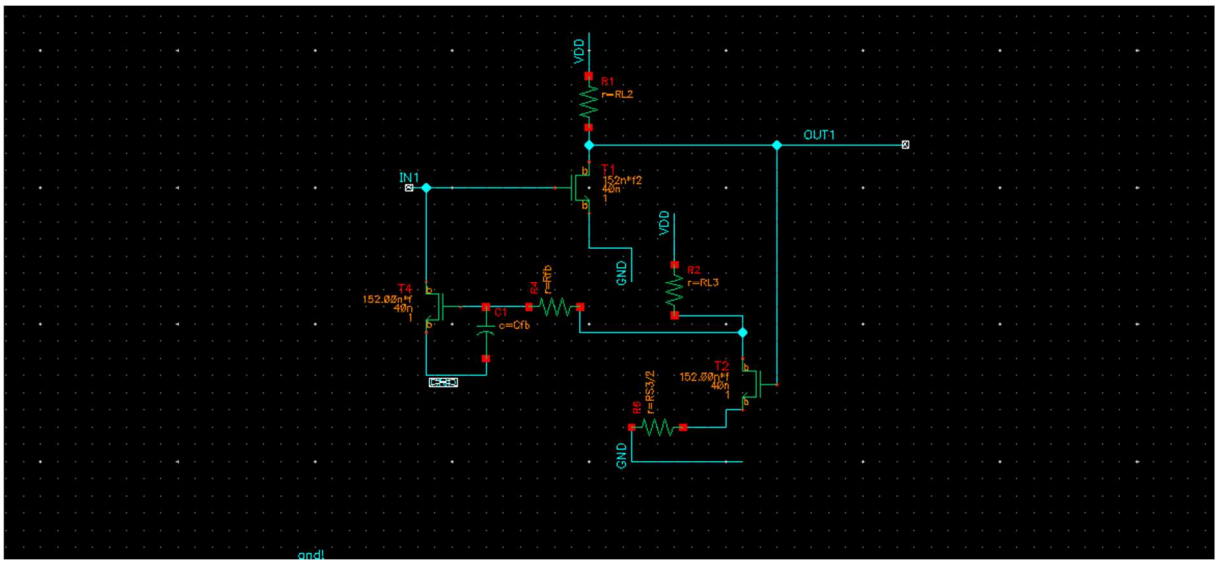
IV-Conclusion

We first characterized our infrastructure with Matlab. After assessing, the different parameters and compromise possible, we chose a specific infrastructure for the receiver. This choice of design enabled us to reach a speed of 25Gb/s. We then divided this infrastructure into different blocks. Each one of us worked on a specific part. With the constrains imposed by the rest of the circuit, I chose a specific infrastructure for the CTLE, made of 2 equalizers and 2 buffers. This CTLE allow us to amplify the high frequencies so that we can detect the incoming data at a speed of 25GB/s. Finally, I designed the layout of this circuit and we hooked up the different layout of the different bloc of this overall circuit. Thus, we were able to provide an operating design of a circuit implemented at the receiver that can recover the quality of the original signal travelling at a speed of 25Gb/s.

Annex

Closed-loop gain for the LFEQ

The differential half circuit is:



Half circuit for the differential mode

To compute the gain, we will use the return ratio method. If we put a voltage test source vt at the input of T1, the voltage v_r is:

$$v_r = \frac{-gm_4.Rl_1.gm_3.Rl_3.(1+\frac{gm_3Rs_3}{2})(1+sC_{fb}.R_{fb})gm_2.Rl_2}{(1+sC_{fb}.R_{fb})(1+\frac{gm_3.Rs_3}{2})(1+sC_{fb}.(Rl_3+R_{fb}))(1+s\,Cl\,Rl_2)}\,v_t$$

So, the return ratio is:

$$RR = \frac{-v_r}{v_t} = \frac{gm_4.Rl_1.gm_3.Rl_3.(1+\frac{gm_3Rs_3}{2})(1+sC_{fb}.R_{fb})gm_2.Rl_2}{(1+sC_{fb}.R_{fb})(1+\frac{gm_3.Rs_3}{2})(1+sC_{fb}.(Rl_3+R_{fb}))(1+s\,Cl\,Rl_2)}$$

Let us compute the feedforward factor. If gm = 0 for T1, then, Vout = 0. So, the feedforward gain d is 0.

Let us compute A_∞. So, if gm -> ∞, then :

$$v_o = \frac{-(1+\frac{gm_3.Rs_3}{2})(1+sC_{fb}.(Rl_3+R_{fb}))}{gm_4.gm_3.Rl_3}\,i_{in}$$

So:

$$A_\infty = \frac{-(1+\frac{gm_3.Rs_3}{2})(1+sC_{fb}.(Rl_3+R_{fb}))}{gm_4.gm_3.Rl_3}$$

Finally, the closed-loop gain is:

$$A_{cl} = \frac{A_\infty \, RR}{1 + RR} + \frac{d}{1 + RR}$$

Thus:

$$A_{cl} = \frac{-gm_2.Rl_1.Rl_2.(1 + \frac{gm_3 Rs_3}{2})(1 + sC_{fb}.(Rl_3 + R_{fb}))}{gm_4.Rl_1.gm_3.Rl_3.gm_2.Rl_2 + (1 + \frac{gm_3.Rs_3}{2})(1 + sC_{fb}.(Rl_3 + R_{fb}))(1 + s\ Cl\ Rl_2)}$$

References

Vladimir Stojanovic, September 2004, CHANNEL-LIMITED HIGH-SPEED LINKS: MODELING, ANALYSIS AND DESIGN

Samir Parikh, Tony Kao, Yasuo Hidaka, Jian Jiang, Asako Toda, Scott Mcleod, William Walker, Yochi Koyanagi, Toshiyuki Shibuya, Jun Yamada, A 32Gb/s Wireline Receiver with a Low-Frequency Equalizer, CTLE and 2-Tap DFE in 28nm CMOS, 2013

Taylor Palmer, IBISWorld Industry Report 51913a Search Engines in the US, September 2016

Darryle Ulama IBISWorld Industry Report 33441b Circuit Board & Electronic Component Manufacturing in the US, June 2016