

IQ modulator based on C-DAC topology for 2GHz transmitter

By Adar Cohen and Alon Bechor, Tel Aviv University

Table of Contents

Introduction	2
Switch Capacitor DAC	3
The RLC circuit.....	7
RF Switch Capacitor DAC Power Amplifier	10
Differential SCPA.....	13
QAM - IQ Modulation	16
Non-Linear Effects.....	23
Resonance	25
Matching Network.....	26
The Smith Chart	28
Standard Cells	34
Load Pull	35
Non-Ideal Transformers.....	40
Matching using Non-Ideal Transformer	42
Matching using Non-Ideal Transformer – Two channels.....	47
256 bit Differential IQ SCPA.....	52
Sizing Logic Gates.....	57
Sizing the Unit Cell	61
Clock Tree.....	64
Data Buffer and Mux	68
Q-SCPA	70
Momentum	76
Q-SCPA with Momentum Matching network.....	81
IQ imbalance	87

Introduction

RF circuits, or radio frequency circuits, are electronic circuits designed to operate at radio frequencies (RF). Radio frequencies refer to the range of electromagnetic frequencies between audio frequencies and microwaves, typically spanning from 30kHz to 300GHz . RF circuits are essential components in various electronic systems, including communication devices, wireless systems, radar systems, and more.

The constant demand for high quality communication, resulted communication companies to use more digital RF circuits, over analog RF circuits. Digital RF circuits use LOGIC gates to generate Analog signals and digitally control the RF circuit. Digital RF circuits are smaller than the typical analog RF circuits, thus having better power consumption. Moreover, they are easier to integrate with additional VLSI circuitry and can even achieve results better than most analog RF circuits.

Wireless communication devices usually communicate using High frequency based signals. Higher frequency bands allow small antennas, offer a larger amount of available spectrum for communication, are less congested and experience lower levels of interference compared to lower frequencies. Thus, when transmitting data, a transmitter usually up converts the data's baseband signal to a high transmission frequency, using a carrier signal. In order to transmit binary data, the transmitter encodes the data by modulating the carrier signal. Some forms of modulations are Amplitude modulation, Frequency modulation and Quadrature Amplitude modulation.

The output stage of a transmitter is the power amplifier (PA). The main functionality of the PA is to drive power into an antenna. Antenna is a radiating element – converts electrical signals into electromagnetic waves for transmission and reception of RF signals. It requires a certain amount of power to be driven. Additionally, for good efficiency, the antenna dimensions are the same as the signal's wavelength $\lambda = c/f$. Thus, a small antenna requires high frequency signals.

Typically, the transmitter generates a signal with low power, which is not sufficient to achieve the range to go from the transmitter to the receiver. The PA solves this issue by providing signal gain. Power is a function of impedance, so additional goal of the PA is to be an interface for the right impedance, from the perspective of the inner circuitry, such that the maximum possible power will be inputted to the PA.

We want high power from the PA, as it determines the operational range of the system. Moreover, we require good power efficiency, as it determines the battery life in mobile systems and the generated thermal heat.

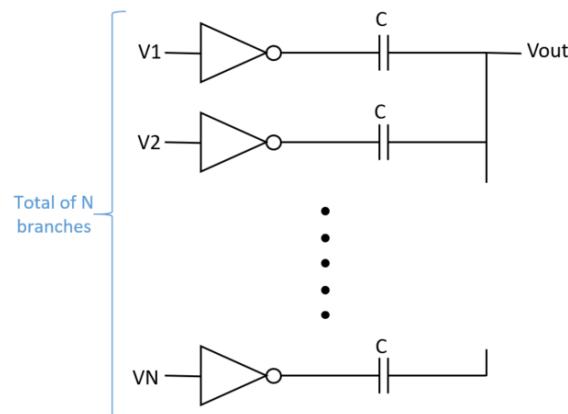
In this project, we implement RF DAC PA, for 2GHz in 65nm technology. The modulation type is IQ QAM. Our goal is to reach 200MHz bandwidth and maximum output power of 20dBm .

Switch Capacitor DAC

Digital to analog converter (DAC) is a circuit that converts digital signals to analog ones, or precisely N bit digital word to an analog voltage.

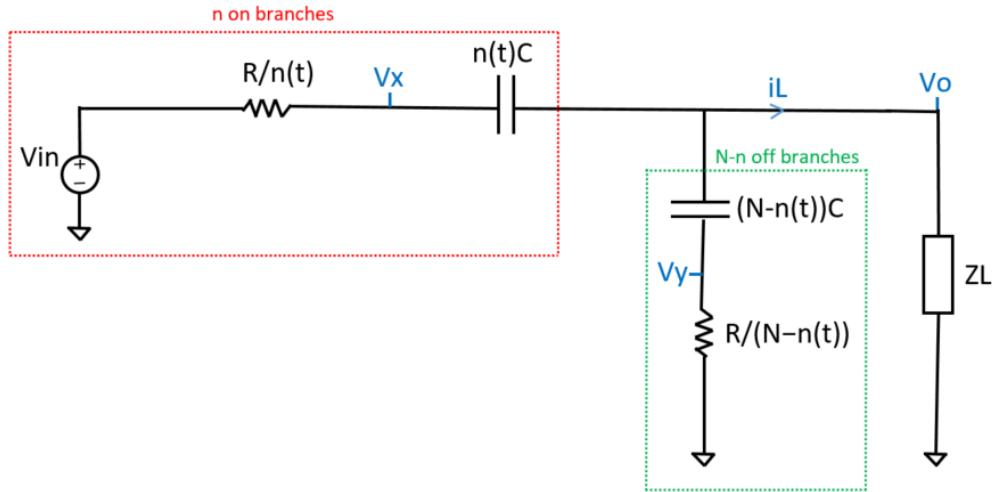
In our design, we use a switch capacitor (SC) DAC, where inverters drive capacitor-combining network. Switch capacitor DAC translates bit code into an analog voltage using capacitors voltage divider. It consists of an array of capacitors whose top plates are shared and whose bottom plates are connected to inverters, which can be switched between the supply voltage and ground. When an inverter's input is high, it switches to ground and when it is low, it switches to the supply voltage. Decoder can selectively switch the inverters, in respect to the input bit code. The output amplitude can thus be modulated, by controlling the ratio of capacitors that are switched to the supply voltage and ground.

Switch Capacitor DAC – Schematic



In the figure above, we can see the design of a switch capacitor DAC. The input voltages are controlled by a decoder and are either V_{dd} or $0V$. These are also the voltage supplies of the inverters. In our design we use thermometric coding (unary), meaning each bit has a weight of one, unlike binary code where each bit has a binary weighing of powers to two. This means that if we have N branches then the input is a N bits word. We denote the number of branches whose inverters are switched to the supply voltage, at some time t , as $n(t)$. We call these $n(t)$ branches on and the rest of the branches off. Additionally, we assume the inverters' transistors have an effective resistance of R and denote the circuit's load as Z_L . Using that, we model the circuit using passive components.

The modeled circuit



Capacitor's current-voltage relationship is $i_c = \frac{d}{dt}(v_c C)$, so:

$$\frac{v_{in} - v_x}{\frac{R}{n}} = \frac{d}{dt}((v_x - v_o)nC) = i_L + \frac{v_y}{\frac{R}{N-n}}, \text{ using KCL on } v_o \text{ junction.}$$

We multiply by R and bring the constant C out of the derivative:

$$\text{Equation 1: } RC \frac{d}{dt}((v_x - v_o)n) = i_L R + v_y (N - n)$$

$$\text{Equation 2: } n(v_{in} - v_x) = i_L R + v_y (N - n)$$

$$\text{From equation 2 we get } v_y = \frac{n(v_{in} - v_x) - i_L R}{N - n}$$

$$\text{We also know that } \frac{v_y}{\frac{R}{N-n}} = \frac{d}{dt}((v_o - v_y)(N - n)C).$$

$$\text{Equation 3: } v_y (N - n) = RC \frac{d}{dt}((v_o - v_y)(N - n))$$

By using equation 3 in equation 1 and using v_y from above, we derive:

$$RC \frac{d}{dt}((v_x - v_o)n) = i_L R + RC \frac{d}{dt} \left(\left(v_o - \frac{n(v_{in} - v_x) - i_L R}{N - n} \right) (N - n) \right)$$

$$C \frac{d}{dt}(v_x n) - C \frac{d}{dt}(v_o n) = i_L + NC \frac{d}{dt} v_o - C \frac{d}{dt}(v_o n) - C \frac{d}{dt}(v_{in} n) + C \frac{d}{dt}(v_x n) + CR \frac{d}{dt} i_L$$

$$i_L + NC \frac{d}{dt} v_o - C \frac{d}{dt}(v_{in} n) + CR \frac{d}{dt} i_L = 0$$

Using Laplace Transform we get:

$$I_L + NCSV_o - CS * L\{v_{in}n\} + CRS * I_L = 0.$$

Using the fact that $Z_L = \frac{V_o}{I_L}$ we get $NCSV_o - CS * L\{v_{in}n\} + \frac{V_o}{Z_L}(1 + CRS) = 0$, so

$$V_o = CS * \frac{L\{v_{in}n\}}{NZ_L + \frac{1}{CS} + R} = \frac{Z_L}{NZ_L + \frac{1}{CS} + R} L\{v_{in}n\} = \frac{Z_L}{Z_L + \frac{1}{N(CS)} + R} L\{v_{in}\tilde{n}\}, \text{ where } \tilde{n} = \frac{n}{N}.$$

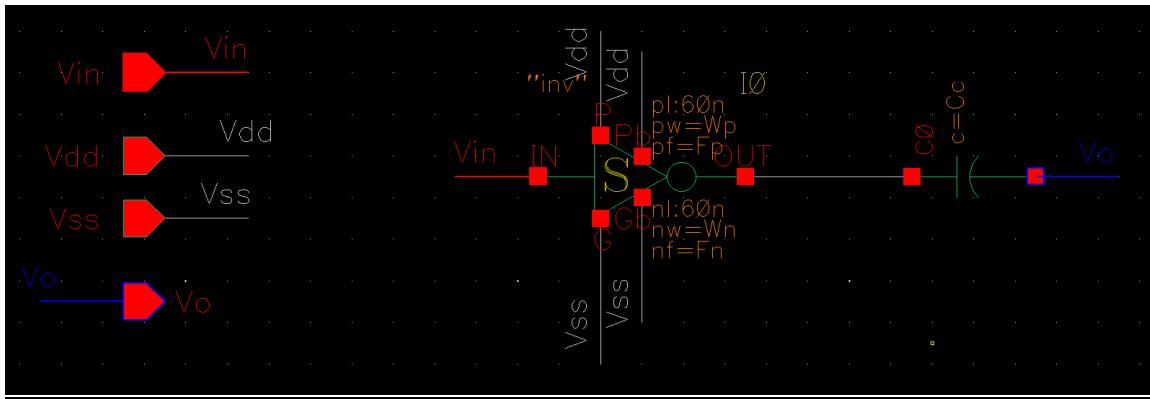
If we denote $Z_o = \frac{1}{N} \left(\frac{1}{CS} + R \right)$ we get:

$$V_o = \frac{Z_L}{Z_L + Z_o} L\{v_{in}\tilde{n}\}$$

If there is no load connected ($Z_L \rightarrow \infty$), $\frac{V_o}{V_{in}} = \tilde{n} = \frac{n}{N}$. Thus, the output voltage is proportional to the number of on branches and to the inputted bit string-as expected from a thermometric coding DAC.

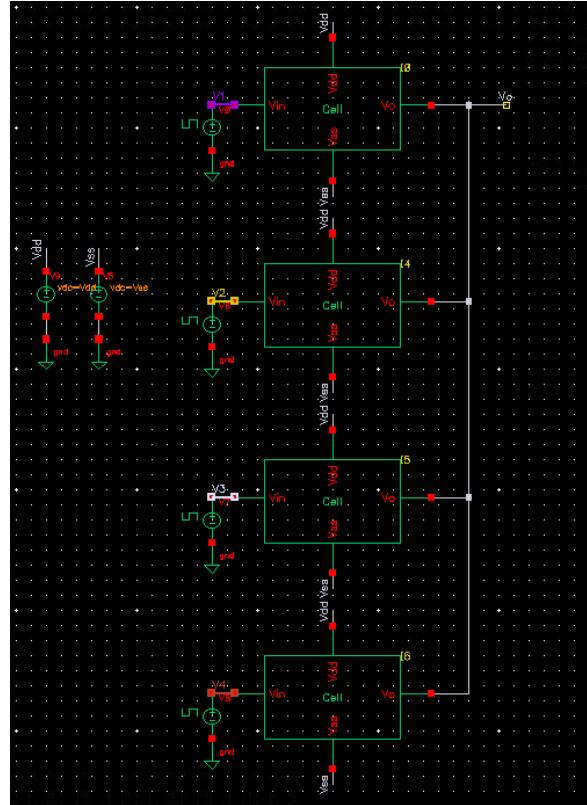
Each branch is also called a cell. We create a cell schematic in virtuoso using a built in inverter and capacitor. The inverter transistors' channel length is $60nm$ and channel width is $200nm$. We control the number of fingers and PMOS has F_p and the number of fingers the NMOS has F_n . Initially $C_c = 150fF$, $F_p = 40$ and $F_n = 20$.

Cell Schematic in virtuoso (a single branch/cell)



We simulate an unloaded 4-bit SC DAC, so the total number of branches/cells is four. We change the input word code and observe how the output changes.

4Bit SC DAC Schematic (no load)

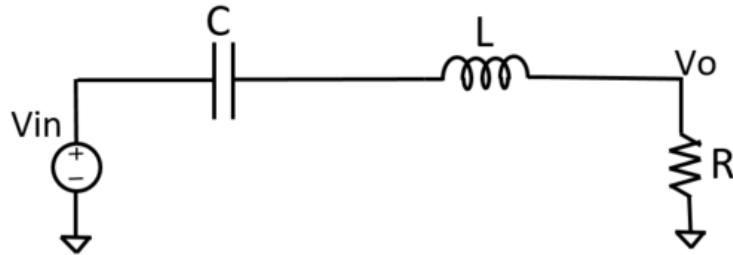


The Output Voltage as a Function of Time and the Bit Code



As expected in the no load case, the output voltage is proportional to the number of on cells and the number of ones in the input word code.

The RLC circuit



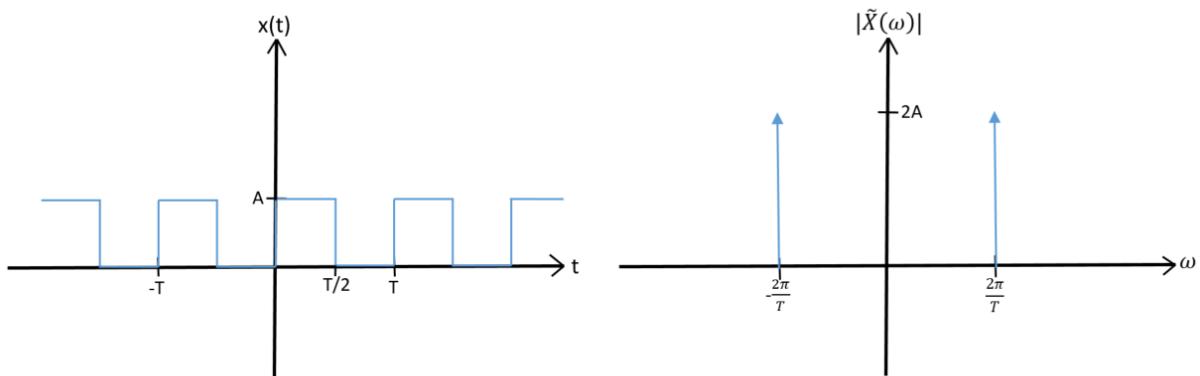
It can easily be derived, from voltage divider, that the above circuit's transfer function is $\frac{V_o}{V_{in}} = \frac{R}{\frac{1}{sC} + sL + R}$. In both low frequencies ($s \rightarrow 0$) and high frequencies ($s \rightarrow \infty$) the transfer function is $\frac{V_o}{V_{in}} = 0$, which means it is a band pass filter.

The input signal to our RLC network is a periodic rectangular wave with 50% duty cycle. We denote its amplitude as A and its period as T. We know that the Fourier Transform (FT) of a periodic signal is $X(\omega) = 2\pi \sum_{k=-\infty}^{\infty} a_k \delta(\omega - \frac{2\pi}{T} k)$, where a_k are the Fourier coefficients and $\delta(\omega)$ is Dirac's delta function. The Fourier coefficients of our input signal are $a_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-j\frac{2\pi}{T} kt} dt = \frac{1}{T} \int_0^{T/2} A e^{-j\frac{2\pi}{T} kt} dt = \frac{1}{T} \left(-\frac{T}{j2\pi k} \right) A e^{-j\frac{2\pi}{T} kt} \Big|_0^{T/2} = -\frac{A}{j2\pi k} ((-1)^k - 1)$.

So, if we pass our signal through a band pass filter, which leaves us only with our fundamental frequency ($\frac{2\pi}{T}$, $k = \pm 1$), we are left with:

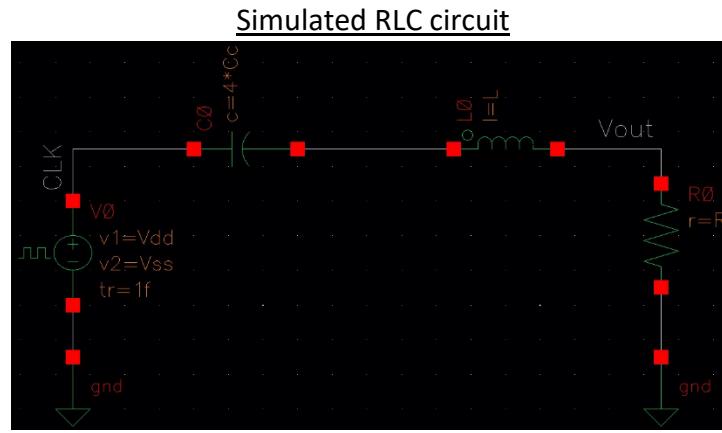
$\tilde{X}(\omega) = 2\pi \left(-\frac{2A}{j2\pi} \delta\left(\omega + \frac{2\pi}{T}\right) + \frac{2A}{j2\pi} \delta\left(\omega - \frac{2\pi}{T}\right) \right)$. Using the fact that $FT\{e^{j\omega_0 t}\} = 2\pi \delta(\omega - \omega_0)$, we get $\tilde{x}(t) = -\frac{A}{j\pi} e^{-j\frac{2\pi}{T}t} + \frac{A}{j\pi} e^{j\frac{2\pi}{T}t} = \frac{2A}{\pi} \sin\left(\frac{2\pi}{T}t\right)$.

The Input Signal (left) and absolute FT of the Filtered Signal (Right)

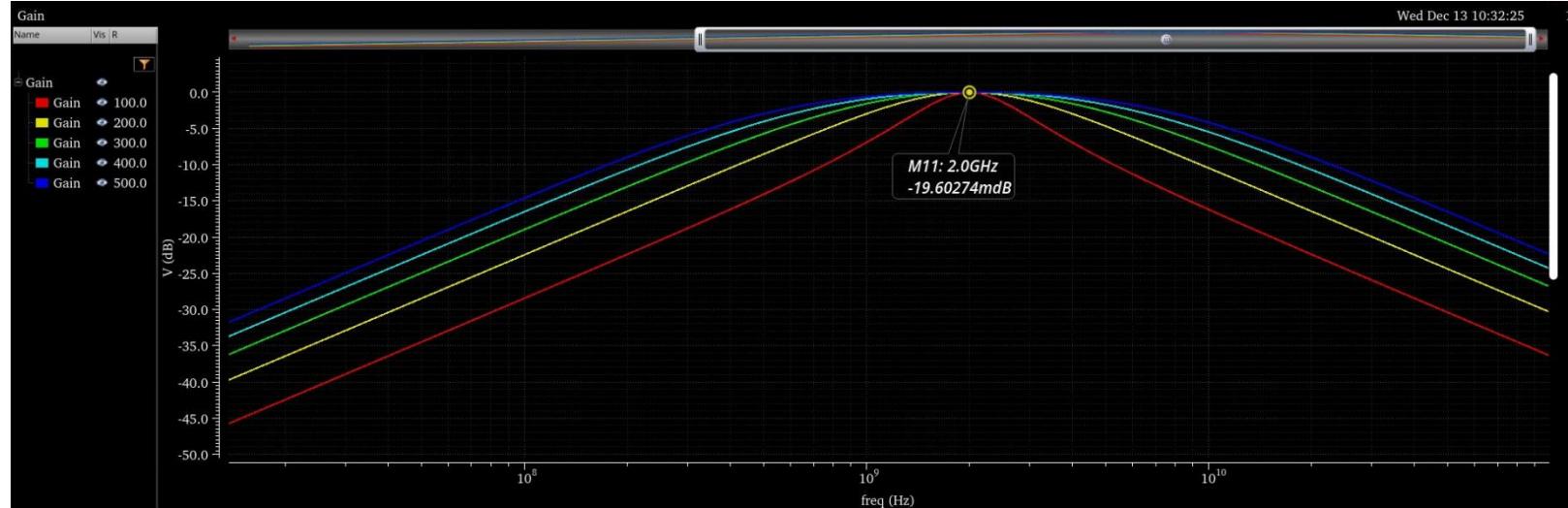


In reality, the bandpass filter is not ideal and does not eliminate all of the non-fundamental harmonics. The BPF transfer function is $H(s) = \frac{V_o}{V_{in}} = \frac{sRC}{1+RCS+CLS^2}$, with poles at $\omega_{p1,2} = \frac{-RC \pm \sqrt{(RC)^2 - 4CL}}{2CL}$. As will be shown later, for maximum power transfer at frequency $\omega_o = 2\pi f_o$, $L = \frac{1}{\omega_o^2 C}$. This leads to the relation $RC = \frac{\omega_{p1} + \omega_{p2}}{\omega_o^2}$. For the poles to not be complex $\omega_{p1} + \omega_{p2} \geq 2\omega_o$.

We simulate an RLC circuit, whose input is a 1.2V periodic rectangular wave with frequency 2GHz. The components parameters are $C = 600fF$, $L = \frac{1}{\omega_o^2 C} = 10.55nH$ and R varies.

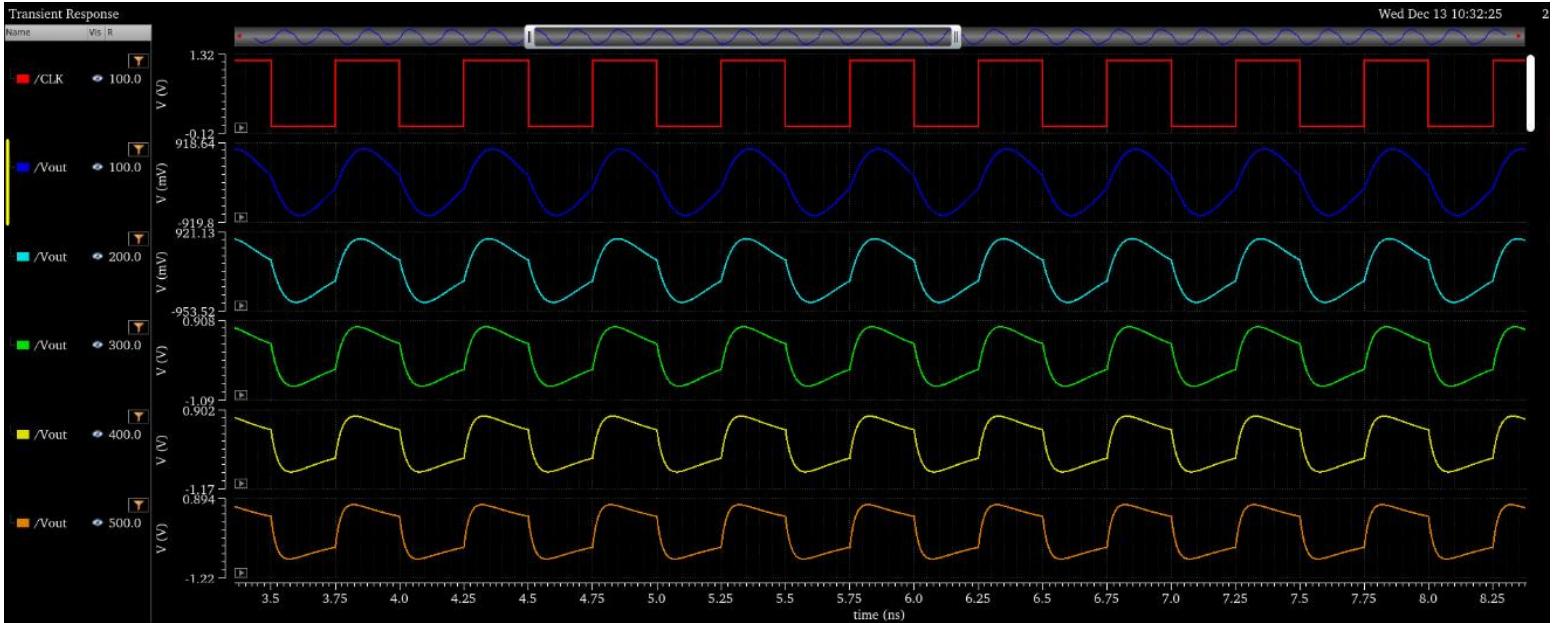


Bode gain graphs



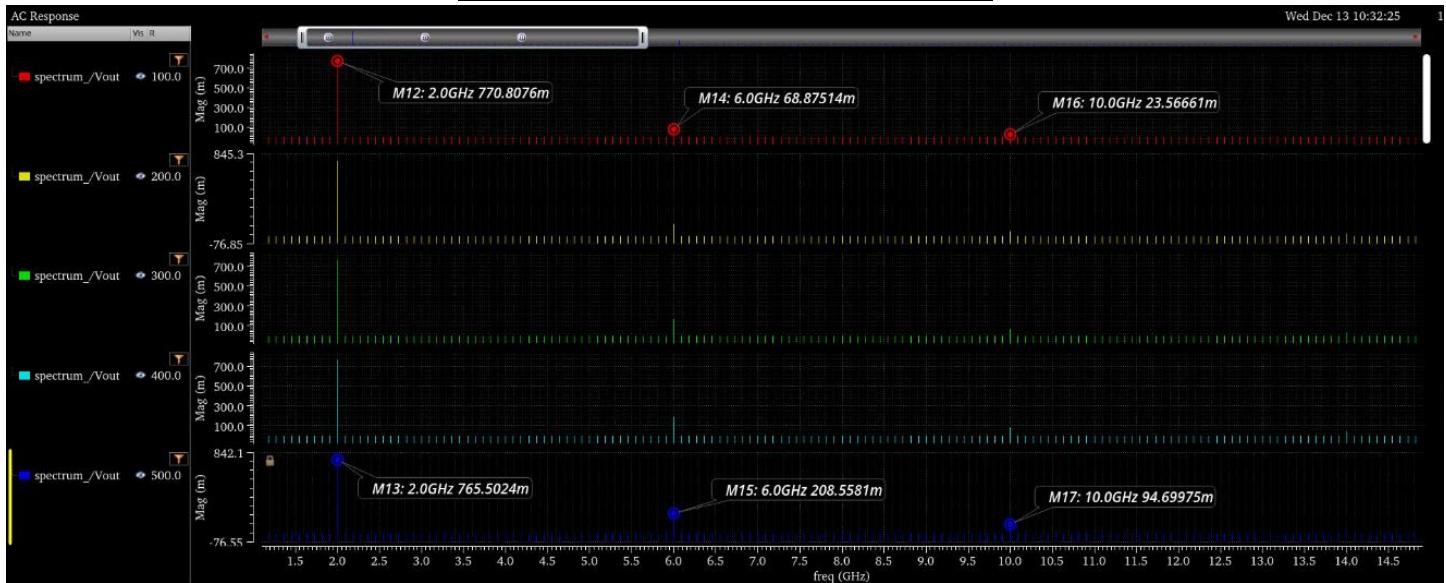
We simulate the Bode gain graph $|H(j2\pi f)|$ for different R values, as shown to the left. The red and yellow graph have complex poles. We see that the smaller R , the narrower the bandwidth. Additionally, the graphs peak at f_o , because we picked $L = \frac{1}{\omega_o^2 C}$.

Output signal for different R values



The red graph is the input signal, the others are outputs for different R values. We see that as R is smaller, the more sinusoidal the output. This coincides with the smaller R the narrower the bandwidth and better the filtering.

Output signal for different R values – FFT



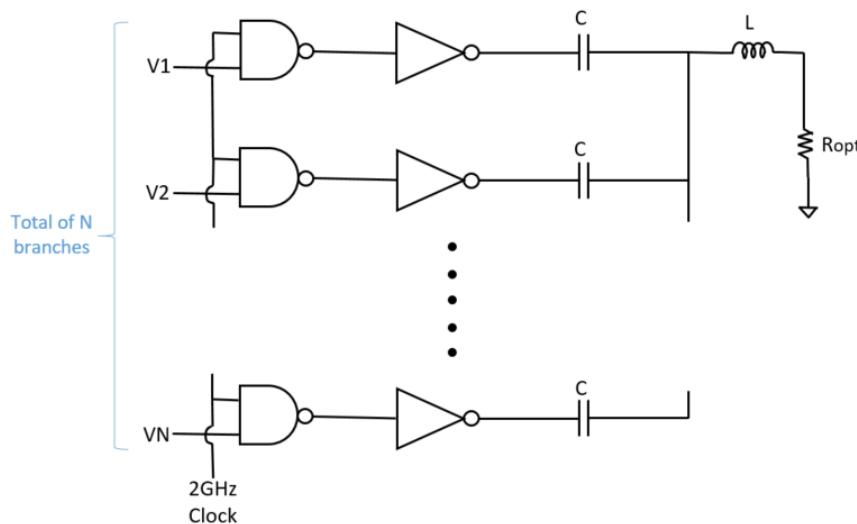
We can look at the FFT of the signals to see their distortion. As expected the non-fundamental harmonics are smaller as R is smaller. From previous calculations, we saw that the fundamental harmonic should have amplitude $\frac{2A}{\pi}$, which is about 0.76V in this case.

RF Switch Capacitor DAC Power Amplifier

In our project we implement the output stage of a semi digital RF transmitter, namely the power stage. In the past, the power stage in wireless transmitter was made of pure analog circuits such as A/AB/B/C amplifiers. Lately there is a trend to implement the output stage using RF-DAC. RF-DAC is a circuit that combines the operation of a mixer (which combines the data signal with a carrier signal) and a DAC. The advantages of using RF-DAC are better immunity to noise and the ability to synthesize multiband (the ability to operate across multiple frequency bands). As a result, the performance of the transmitter is improved, while its cost and complexity is decreased.

We use a SC RF-DAC namely SCPA. The SCPA is a class D PA that contains complementary switches (inverters) and LC resonant network. It has fast low loss switches and controlled capacitance ratio. The PA's output power is proportional to the number of switching cells.

Switch Capacitor Power Amplifier – Schematic



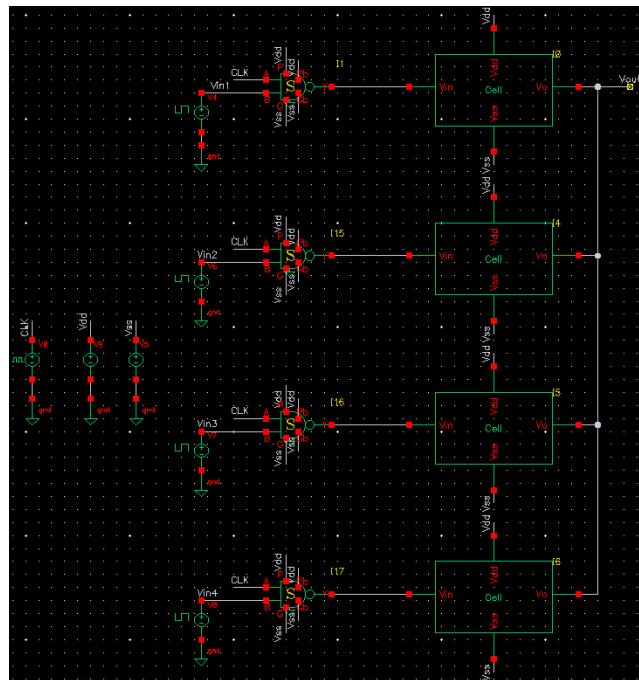
The NAND gates are used as mixers. They mix the input bit code with a 2GHz clock (carrier signal). When a bit is zero, its respective branch is off and its NAND gate outputs a constant one. On the other hand, when a bit is one, its respective branch is on and its NAND gate outputs the invert of the clock.

L is the reactive impedance and R_{opt} is the optimal active impedance of the impedance matching network. They insure minimum signal reflection and the required output power and power efficiency. The inductor is connected in series with the top plates of the capacitors, to filter the square switching waveforms at the SCPAs input. This inductor forms a series resonant circuit with the output resistor; hence it acts as a bandpass filter for the fundamental operation frequency (2GHz). The inductor and output resistor may be formed by passive components, or they can comprise a bandpass

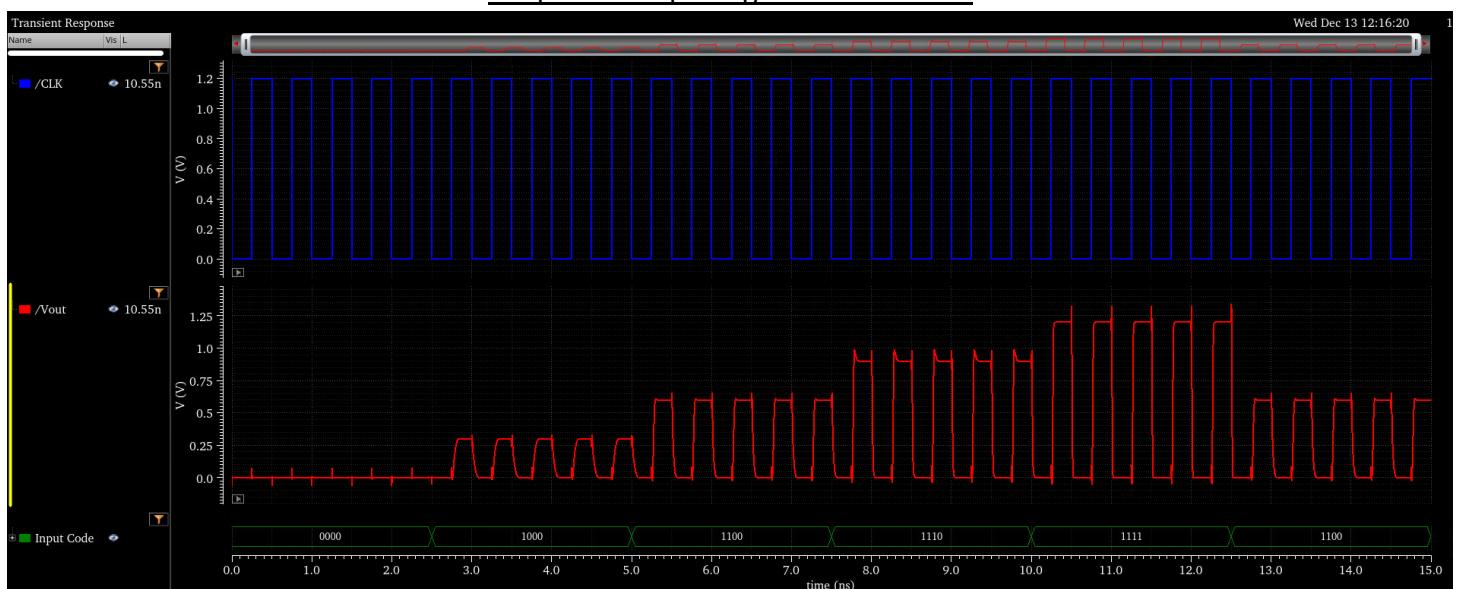
matching network that transforms the impedance of an antenna to an equivalent small resistance in series with a positive reactance.

First of all, we simulate *SCPA*, made of four cells, without matching network. We insert different input codes and observe how does the output changes. The transistors channel length is $60nm$ and their channel width is $200nm$. The PMOS have $F_p = 40$ fingers and the NMOS has $F_n = 20$ fingers. The low voltage supply is $0V$ and the high is $1.2V$. The cells' capacitor is $C_c = 150fF$.

Simulated SCPA (no matching network)



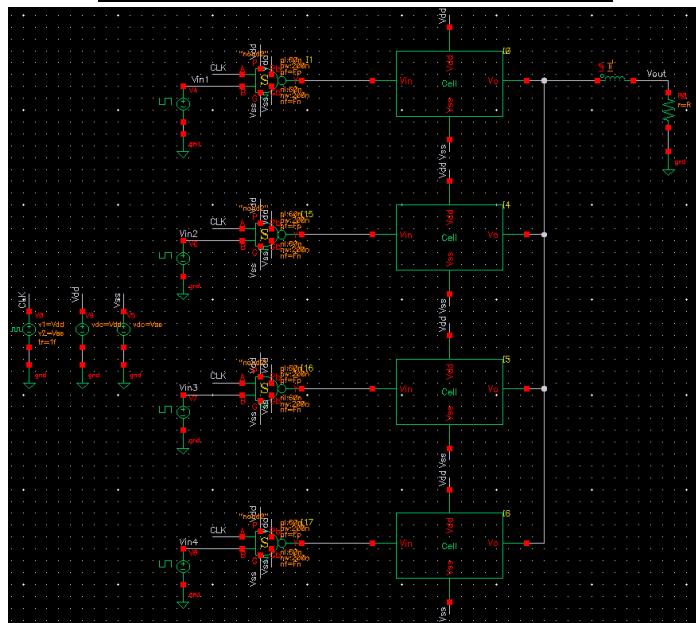
Output and Input Signals versus Time



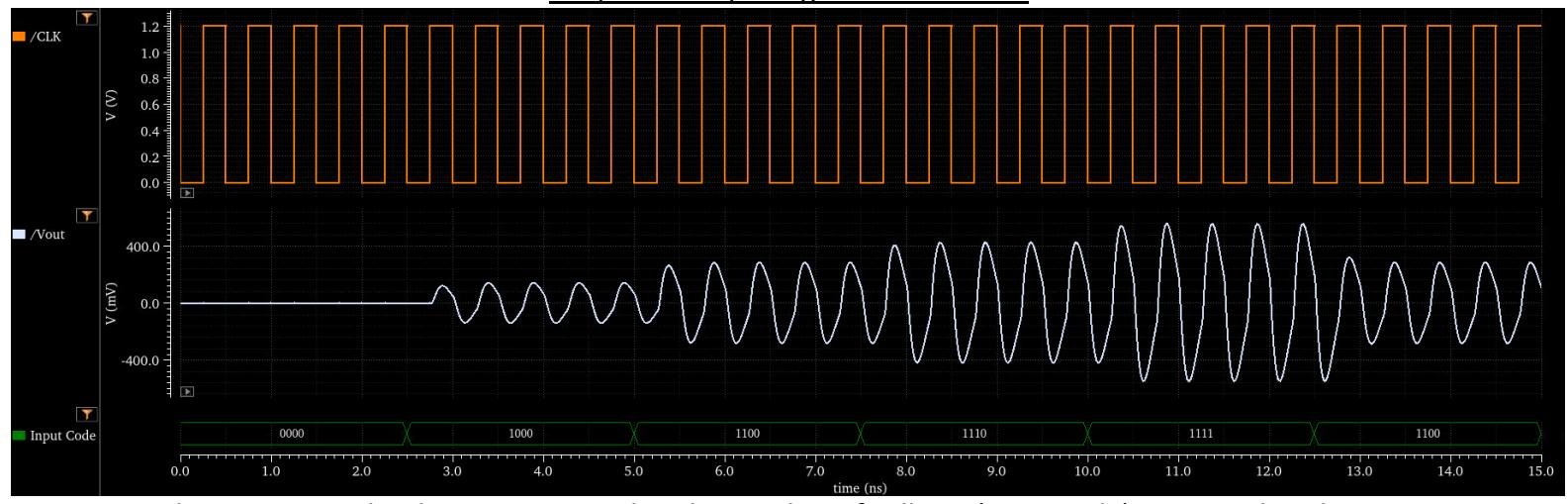
The output amplitude is proportional to the number of cells on (input code), as expected from SC DAC. Because there is no matching network, there is no filtering and the output signal is rectangular, like the input.

Next, we simulate *SCPA* with the matching network. We build the matching network for the case of maximum power transfer, when all of the cells are on. If the cells' inverters are large enough, we can neglect their resistance and treat all of the cells like a single capacitor with capacitance $C = NC_c$, where N is the number of cells. In this case $C = 4C_c = 600fF$. This is the capacitance we used in the RLC simulation. We saw that by taking $L = 10.55nH$ and $R = 100\Omega$ we get fine filtering.

Simulated SCPA with matching network



Output and Input Signals versus Time

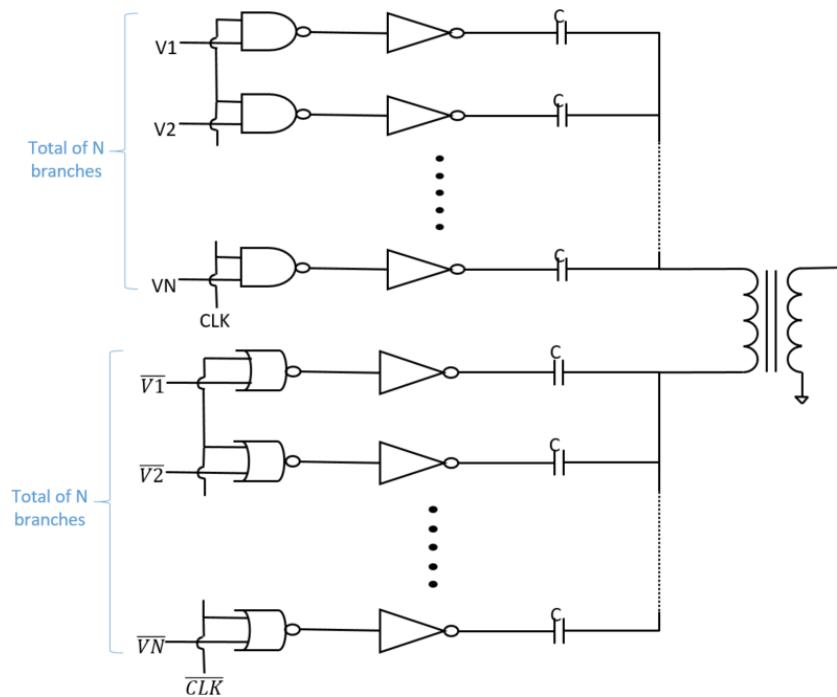


The output amplitude is proportional to the number of cells on (input code). We see that the matching network filtered most of the signals harmonics, so the output signal looks more sinusoidal.

Differential SCPA

The SCPA we use in our project is differential. Differential connections give us better common mode noise rejection. Common mode noise refers to any noise or interference that affects both connection lines equally. In a differential configuration, the noise affecting both lines is canceled out as the circuit only responds to the voltage difference between the lines. Additionally, differential connections have increased dynamic range (output swing), which increases the possible output power. Lastly, differential connections create virtual ground in the circuit. In some circuits, it is a problem to wire ground connection to all of the areas in the chip. Virtual ground creates a reference point, which behaves like a real ground and solves this problem.

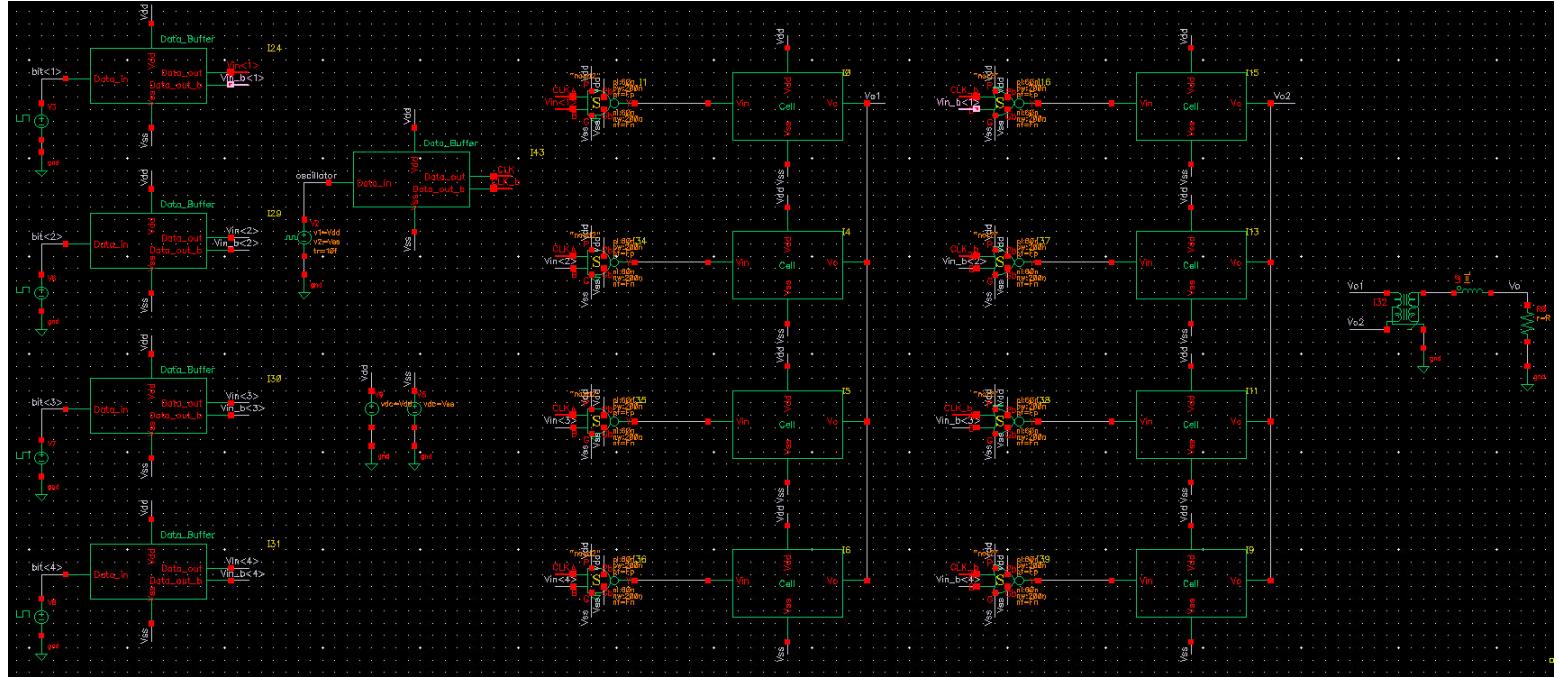
Differential SCPA - Schematic



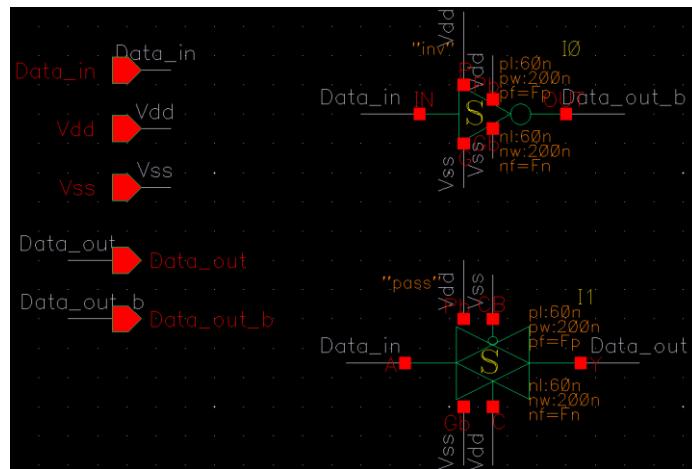
In our design, we obtain differentiability by using two SCPA arrays, like shown in the figure above. The upper array is the one showed previously. The bottom array uses NOR gates as mixers, instead of NAND gates. Its inputs are the inverted input bits and the inverted clock. In this way, in every moment, every array will output minus the output value of the other array. The output of the SCPA is single ended, so in order to move from differential output to single ended one we use a transformer, as shown in the figure.

We simulate four bit differential SCPA with the matching network. We build the matching network for the case of maximum power transfer, when all of the cells are on. The transformer is ideal, so the impedance to the right of the transformer is mirrored perfectly to the right. Because of differentiability and virtual ground, each array sees half this impedance. Thus, we pick twice the inductance and resistance picked in the not differential SCPA simulation, $L = 21.1nH$ and $R = 200\Omega$. To create the inverted input signals and clock we used a buffer.

Simulated Differential SCPA

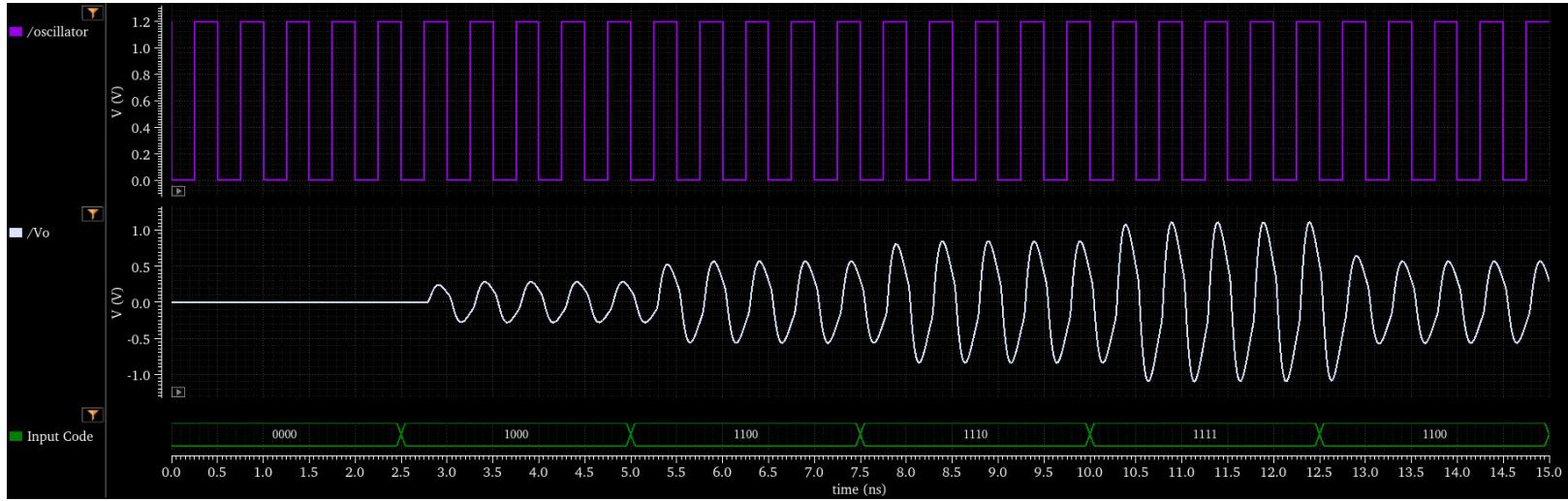


Buffer schematic



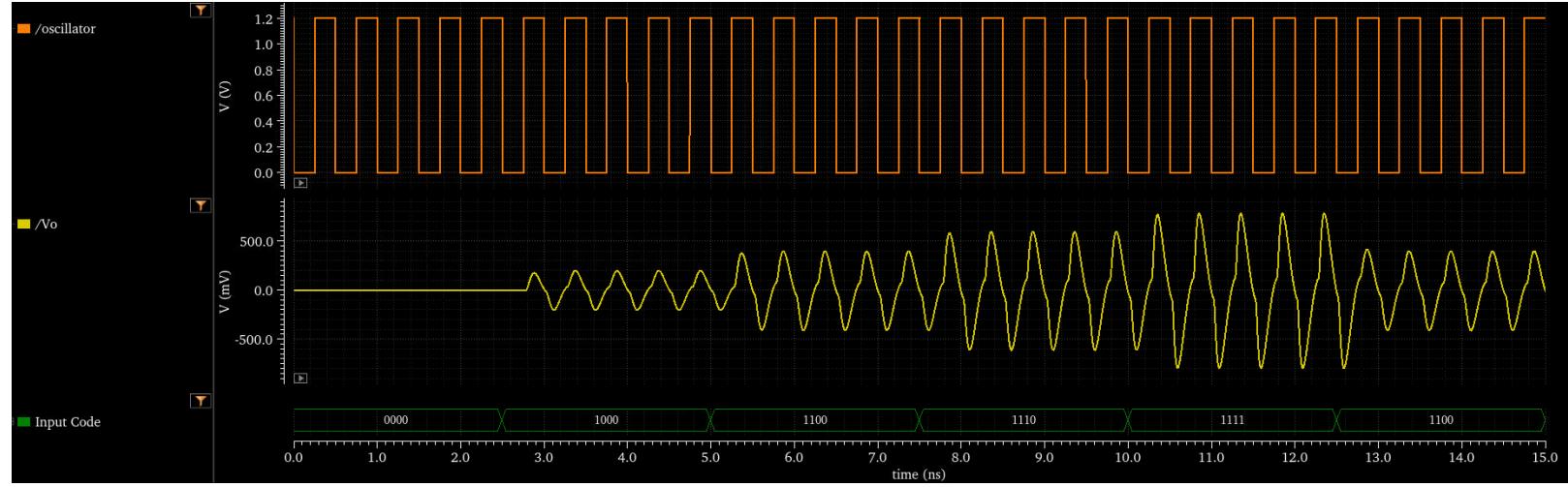
On one hand, the data goes into an inverter, to be inverted. On the other hand, it goes through a TG switch, so the non-inverted signal will go through about the same delay as the inverted signal.

Input and Output Signals versus Time



The output amplitude is proportional to the number of on cells (input code). We see that the matching network filtered most of the signals harmonics, so the output signal looks more sinusoidal. The output signal swing is twice the non-differential SCPA output swing.

Input and Output Signals versus Time – worse matching



This simulation result is for $L = 10.55nH$ and $R = 100\Omega$. As we can see the output signal is more distorted and looks less like a sinusoid. Additionally, the output swing is smaller.

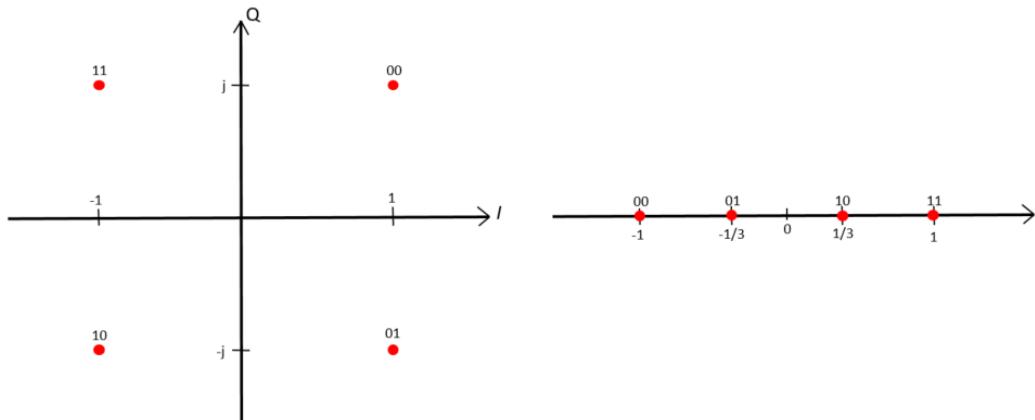
QAM - IQ Modulation

In order to transmit binary data, the transmitter encodes the data by modulating the carrier signal. Some forms of modulations are Amplitude modulation, Frequency modulation and Quadrature Amplitude modulation (QAM). Our design is based on QAM modulation, namely IQ.

The motivation behind using QAM modulation goes as follows. Let's say we want to transmit n bit word using Amplitude modulation, where we control the signal's amplitude. Additionally, suppose our carrier signal's amplitude is bounded between -1V and 1V. Then each word is assigned to a value, called symbol, between this range, where there is a total of 2^n symbols. Because synchronization problems and noise, we want to spread these values evenly, such that the minimal distance between values is maximal. Thus, in this case, the optimal distance is $\frac{2}{2^n-1}$.

By improving the minimal distance, more bits per second (data rate) can be transmitted, since there is less sensitivity to noise. A way to improve the minimal distance is to use complex values. The figure below compares the case where we want to transmit 2 bit words, a total of 4 symbols, using complex values (left graph axes), versus real values (right axis). In the case of complex values the minimal distance between symbols is 2, whereas in the real values case the minimal distance is $\frac{2}{2^2-1} = \frac{2}{3}$.

4QAM IQ modulation vs Amplitude modulation



In order to transmit complex values, QAM modulation can be used. In the QAM modulation complex values can be represented using two Amplitude modulation channels (IQ) or amplitude and phase modulation (polar). In our design we use IQ modulation, which is built out of two channels, I and Q, with orthogonal carrier signals (90deg phase difference clocks with the same frequency). The modulated carrier signals

are summed before transmission. Due to the nature of orthogonal signals the receiver can reconstruct the I and Q components from the received signal.

As discussed above, the advantage of IQ modulation is that it has better resistance to noise and interference. Additionally, it is simpler than the Polar modulation, where additional VLSI circuitry is required. Moreover the symbol transmission rate is lower, as each symbol consists of more bits, hence the bandwidth is smaller. On the other hand, it consumes more power, than both Amplitude and Polar modulations.

In our design, to generate orthogonal I and Q components, we use two differential SCPA with orthogonal clocks (90° phase difference). For correct constellation, the I clock rising edge needs to lead the Q clock rising edge by 90° . To sum up these components we use ideal transformers in series. To move from differential to single ended output we use an additional transformer.

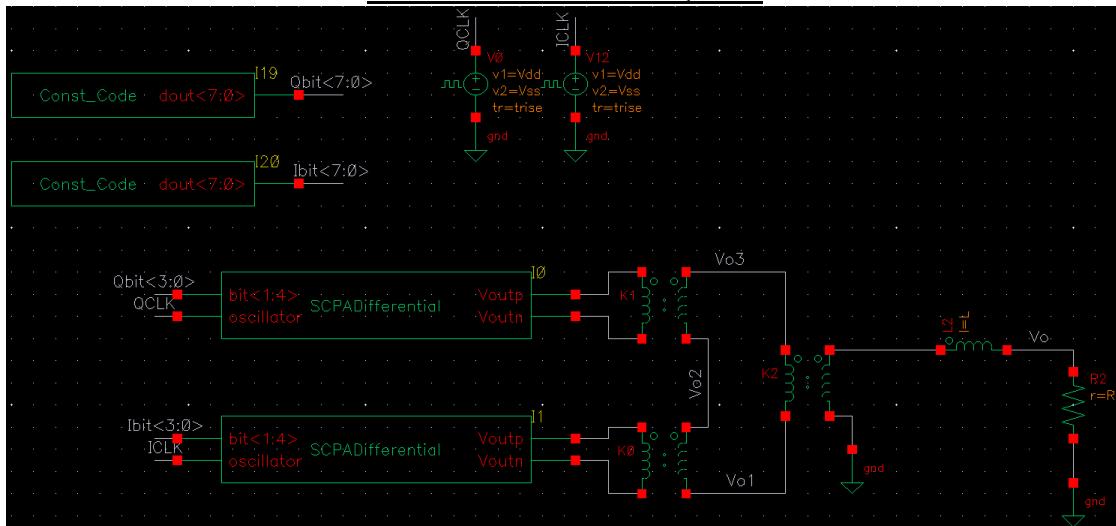
Our design is made to transmit in 2GHz . Hence, we care only about the output voltage fundamental frequency, which is 2GHz . We denote the number of cells on in the Q and I channels as n_Q and n_I , respectively. We denote ϕ_o as the offset phase of the IQ constellation. It is the phase of the output voltage, when only the I channel transmits and only its cells are on. Thus, the output voltage phase should be $\phi V_{out} = \tan^{-1}\left(\frac{n_Q}{n_I}\right) + \phi_o$.

We denote the output voltage amplitude, each cell contributes to its channel's component, as v_o . For example, n_Q on cells should result the output voltage's Q component to have an amplitude of $n_Q \cdot v_o$. Thus, the output voltage amplitude should

$$\text{be } |V_{out}| = v_o \sqrt{n_Q^2 + n_I^2}.$$

We simulate differential IQ SCPA with matching network. Each channel has 4-bit input. We use the same resistance and inductance as before $L = 21.1\text{nH}$ and $R = 200\Omega$.

Simulated Differential IQ SCPA



Cosnt_code is a self-made VerilogA component. It is used to control how many cells are on in each channel. Its parameters are high voltage and "Word code". "Word code" is a positive decimal integer between 0 and 255. It outputs the respected binary value of "Word code", where ones are the high voltage parameter and zeros are zero voltage. In our case the high voltage is 1.2V.

Const_Code VerilogA code

```

`include "constants.vams"
`include "disciplines.vams"

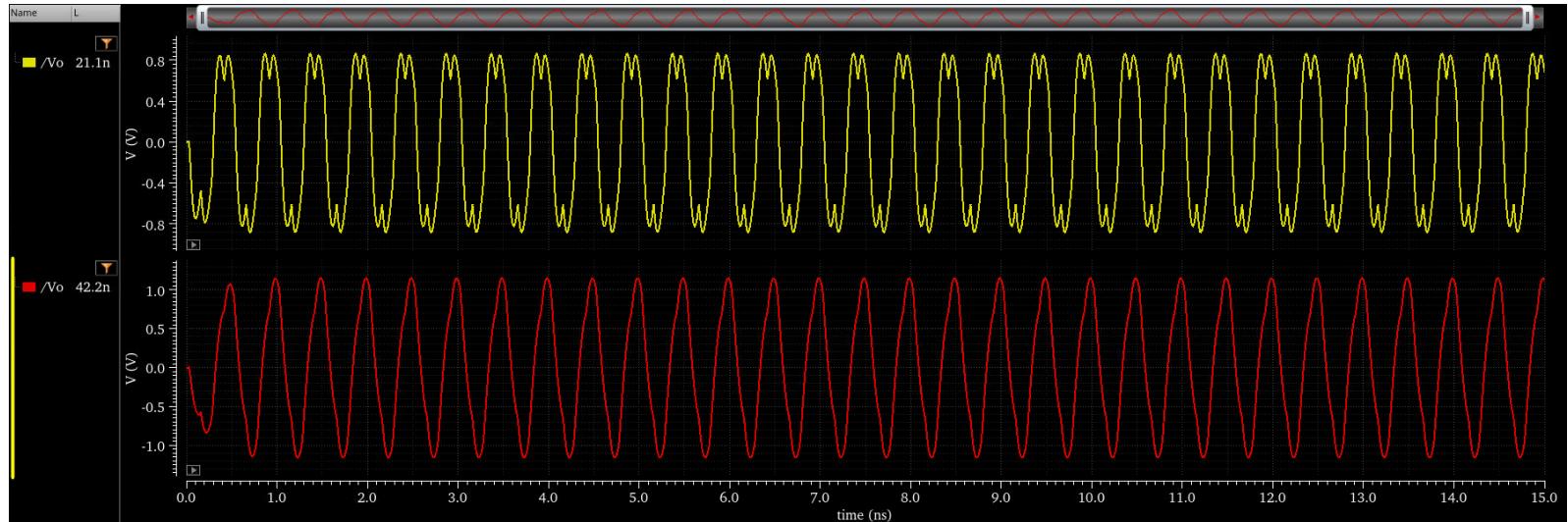
module Constant_Word(dout);
output [7:0] dout;
electrical [7:0] dout;

parameter integer Word_Code = 0 from [0:255];
parameter real vlogic_high = 1.2;
parameter real vlogic_low = 0;
real vout[7:0];
integer num;
integer i;
genvar k;

`define INPUT_BITS     8
analog begin
    for (i = 0; i < `INPUT_BITS ; i = i + 1) begin
        vout[i] = (((Word_Code / (2 ** i)) % 2) > 0) * vlogic_high;
    end
    for( k = 0; k < `INPUT_BITS; k = k + 1) begin
        V(dout[k]) <+ vout[k];
    end
`undef INPUT_BITS
end
endmodule

```

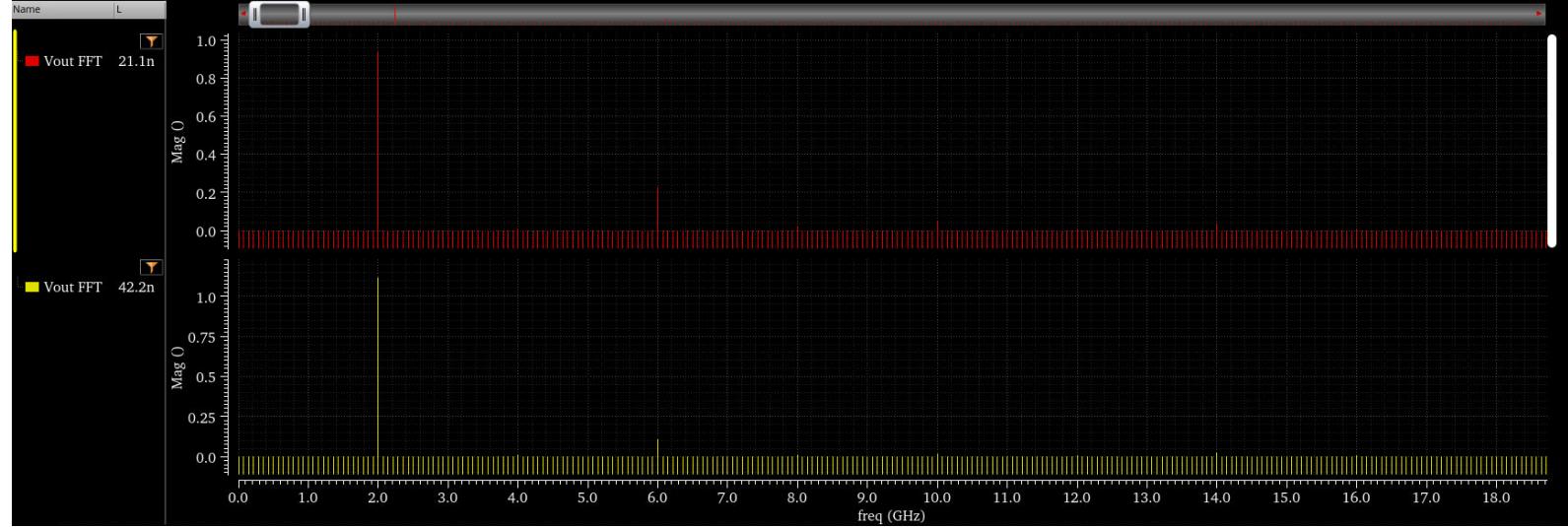
Output voltage when all cells are on for different inductances



The following are the output voltage signals when both I and Q channel are fully on. The

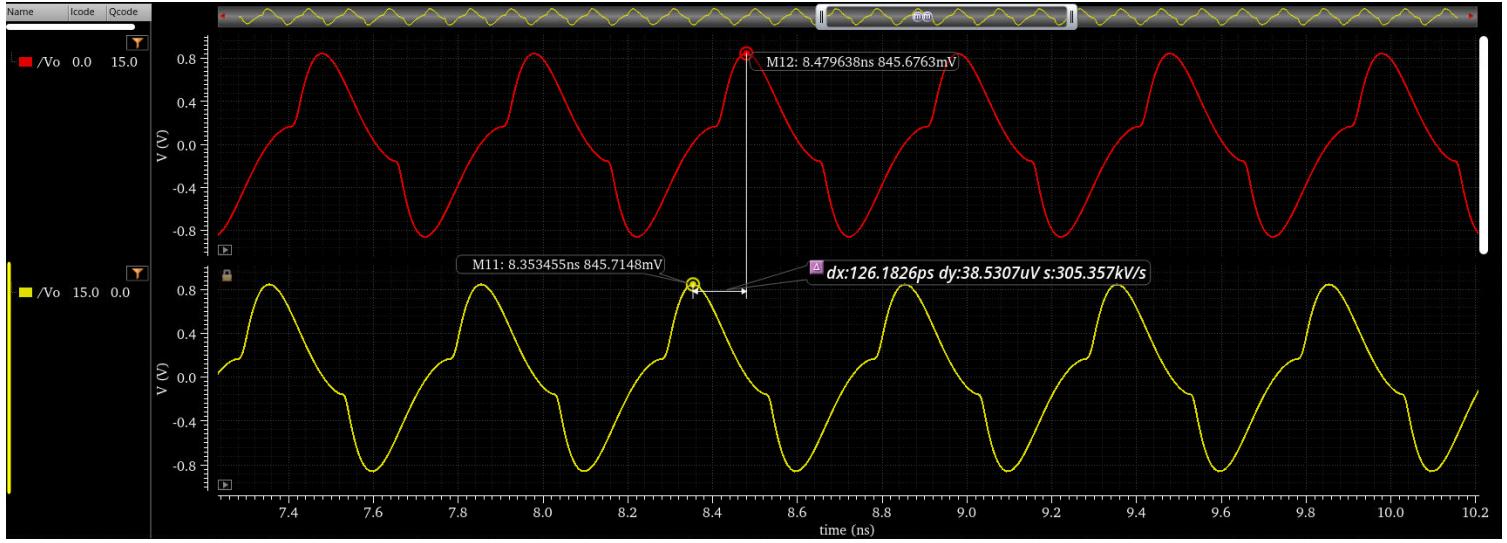
yellow graph is for $L = 21.1nH$ and the red graph is for $L = 42.2nH$. The red graph is much more sinusoid like, which means $L = 42.2nH$ is better for filtering.

Output voltage when all cells are on for different inductances – FFT



By looking at the FFT of the output signals we can observe their distortion levels. Here the colors switched and the red graph is for $L = 21.1nH$. The red graph's non fundamental harmonics are more dominant, hence more distortion. However, in the following simulations we mainly care about transmitting using IQ modulation, so it is fine to take $L = 21.1nH$.

Output voltage when I and Q channels are fully on separately



The red graph is the output voltage signal when only the Q channel is on. The yellow graph is the output when only the I channel is on. Because in each case only one channel is on, we expect the output signals to be orthogonal.

The signals look the same and the phase difference between them is about 0.125ns .
The period of the signals is 0.5ns , so the phase difference is a quarter of a period, which is a 90° difference and means the signals are indeed orthogonal.

Virtuoso used expressions

Vout	signal	<input checked="" type="checkbox"/>	/Vo
QVout	expr	<input checked="" type="checkbox"/>	(VT."/Vo3") - VT."/Vo2"))
IVout	expr	<input checked="" type="checkbox"/>	(VT."/Vo2") - VT."/Vo1"))
Vout FFT	expr	<input checked="" type="checkbox"/>	dft(VT."/Vo") 1e-09 1.5e-08 (2**14) "Rectangular" 1 1 1.0)
Vout magnitude	expr	<input checked="" type="checkbox"/>	mag(dft(VT."/Vo") 1e-09 1.5e-08 (2**14) "Rectangular" 1 1 1.0))
Vout phase	expr	<input checked="" type="checkbox"/>	phaseDeg(dft(VT."/Vo") 1e-09 1.5e-08 (2**14) "Rectangular" 1 1 1.0))
H1 magnitude	expr	<input checked="" type="checkbox"/>	value(mag(dft(VT."/Vo") 1e-09 1.5e-08 (2**14) "Rectangular" 1 1 1.0)) VAR("frf"))
H1 phase	expr	<input checked="" type="checkbox"/>	value(phaseDeg(dft(VT."/Vo") 1e-09 1.5e-08 (2**14) "Rectangular" 1 1 1.0)) VAR("frf"))
H1 Pout	expr	<input checked="" type="checkbox"/>	dBm(0.5 * real(value((dft(VT."/Vo") 1e-09 1.5e-08 (2**14) "Rectangular" 1 1 1.0) * conjugate(dft(IT."/R2/PLUS") 1e-09 1.5e-08 (2**14) "Rectangular" 1 1 1.0))) VAR("frf"))))

Using the following expressions, we extract the phase and amplitude of the fundamental frequency 2GHz . Each channel has five possible states, from none of its cells on to all of its cells on. We sweep over all the states of each channel and extract the phase and magnitude each time. Using Matlab, each phase θ and magnitude r we transform into its I and Q components, $I = r\cos(\theta)$ and $Q = r\sin(\theta)$. We plot all of the results on one graph, to get the IQ constellation diagram.

Matlab code

```

clc
close all

cosDegrees = cosd(H1Phase);
sinDegrees = sind(H1Phase);

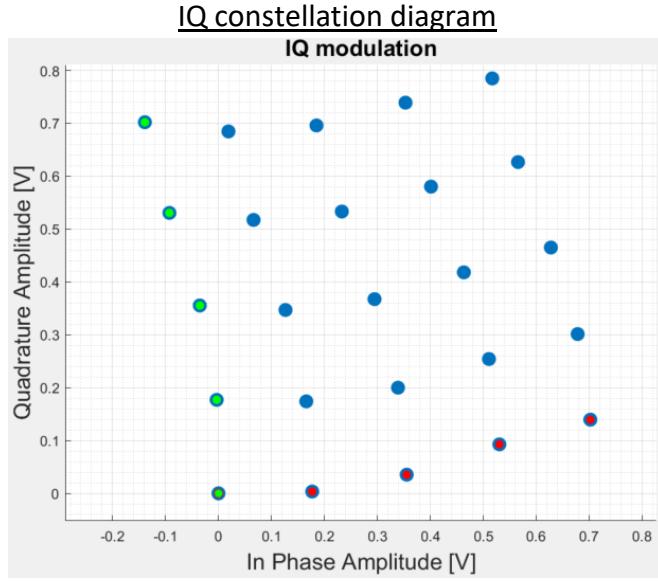
Icomponent = H1Magnitude .* cosDegrees;
Qcomponent = H1Magnitude .* sinDegrees;

scatter(Icomponent, Qcomponent, 100, 'filled');
title('IQ modulation', 'FontSize', 12);
xlabel('In Phase Amplitude [V]', 'FontSize', 12);
ylabel('Quadrature Amplitude [V]', 'FontSize', 12);
grid on
grid minor
hold on

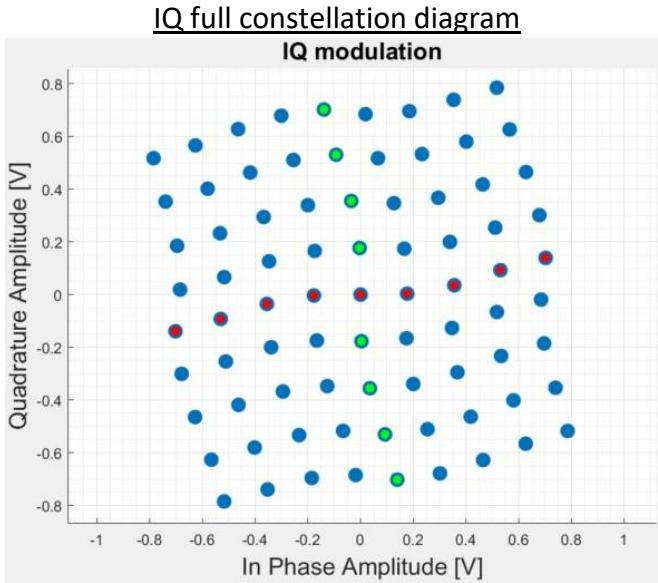
scatter(Icomponent(1:5), Qcomponent(1:5), 'filled', 'red') %I
array is off
scatter(Icomponent(1:5:21), Qcomponent(1:5:21), 'filled', 'green') %Q array is off
scatter(Icomponent(1), Qcomponent(1), 'red')

hold off

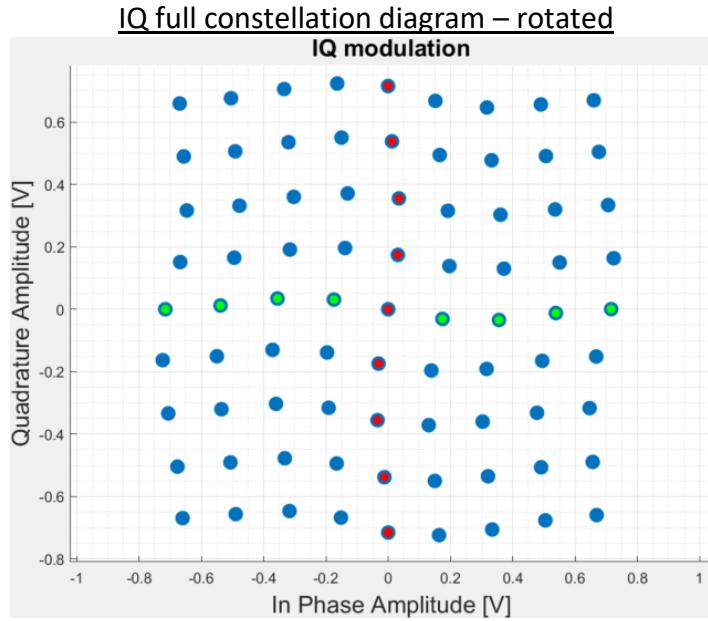
```



The following is the IQ constellation. The red and green points refer to cases where the I and Q arrays (channels) are off, respectively. These points are approximately orthogonal. Our setup lets us transmit data only in quarter of the constellation diagram. We can transmit data in other quarters by inverting the clocks. We do that by using two additional bits. Each bit controls whether a clock should be inverted or not.



By changing the clocks polarity we are able to construct the full IQ constellation, as shown above. As before, the red and green points refer to cases where the I and Q arrays (channels) are off, respectively.



It makes more sense that the red and green points, where only the Q array and I array are on respectively, would be placed on the Q and I axis respectively. Thus, we rotate the constellation plane by 78.8° and get the above diagram.

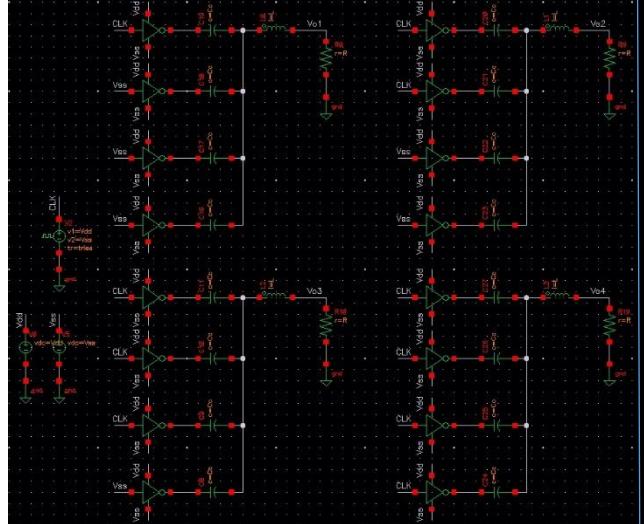
We can see that the constellation is curved. Even when only the I or Q channels are on, the points are not sitting on a straight line, but each has different phase. The output signal's phase depends on how many cells are on, unrelated to the IQ components.

A possible cause is the nonlinear behavior of the PA, phase and amplitude distortions are added to the fundamental frequency. Additional possible cause is IQ imbalance. It results from the I and Q components not being exactly 90° apart, or from unequal gain from the I and Q channels. These result in distorted constellation. However, we don't have a reason to believe this is the case here, because the simulation is ideal.

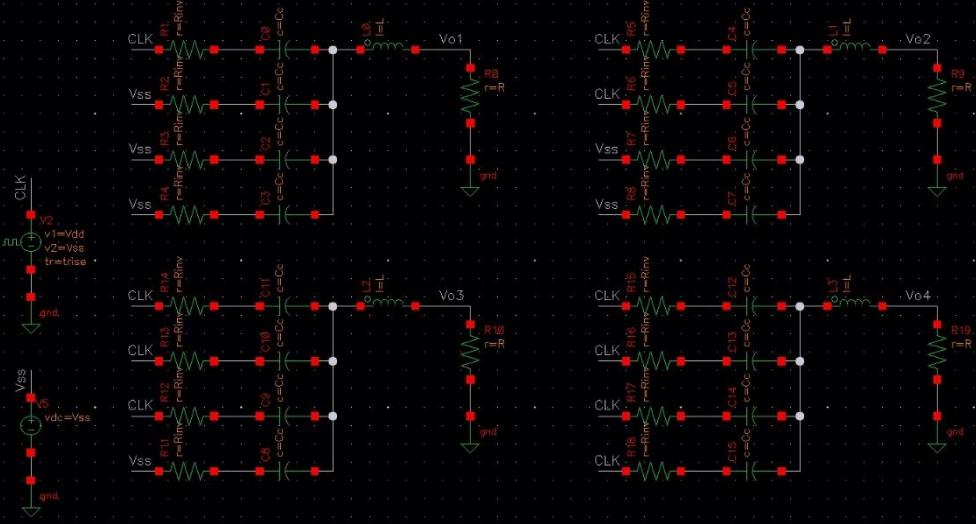
Non-Linear Effects

As we saw, the nonlinearity of the circuit causes distortion in the IQ constellation. To understand the cause of the nonlinearity we simplify the problem. We simulate a 4bit, single channel, non-differential SCPA. In one case, we use inverters to push the capacitors, as done until now, and in the other case we model the inverters as 1Ω resistors.

With inverters case



Without inverters case



In each case, we duplicate the circuit four times. Each of the circuits have a different amount of cells that are on, between one and four. Using simulation, we find the output voltage amplitude and phase, of the fundamental harmonic ($2GHz$).

<u>With inverters results</u>	<u>Without inverters results</u>
Vout1 magnitude	64.52m
Vout2 magnitude	129.4m
Vout3 magnitude	194.3m
Vout4 magnitude	259m
Vout1 phase	-50.03
Vout2 phase	-48.58
Vout3 phase	-47.56
Vout4 phase	-47.22
Vout1 magnitude	67.44m
Vout2 magnitude	134.9m
Vout3 magnitude	202.3m
Vout4 magnitude	269.8m
Vout1 phase	144.9
Vout2 phase	144.9
Vout3 phase	144.9
Vout4 phase	144.9

The number adjacent to $Vout$ is the number of cells that are on. For example, $Vout3$ means that this is the output voltage for the circuit with 3 cells on.

We begin by looking at the case without inverters. The output's amplitude is proportional to the number of on cells $|Vout| = n * 67.44m$ and the phase is constant. In a linear system, at a specific frequency, the output's amplitude is proportional to the input's amplitude, $|Y(j\omega_o)| = |H(j\omega_o)||X(j\omega_o)|$. In this case, the input's amplitude is a discrete number n . Additionally, a linear system, at a specific frequency, adds a constant phase to the input $\angle Y(j\omega_o) = \angle H(j\omega_o) + \angle X(j\omega_o)$. This indicates that the case

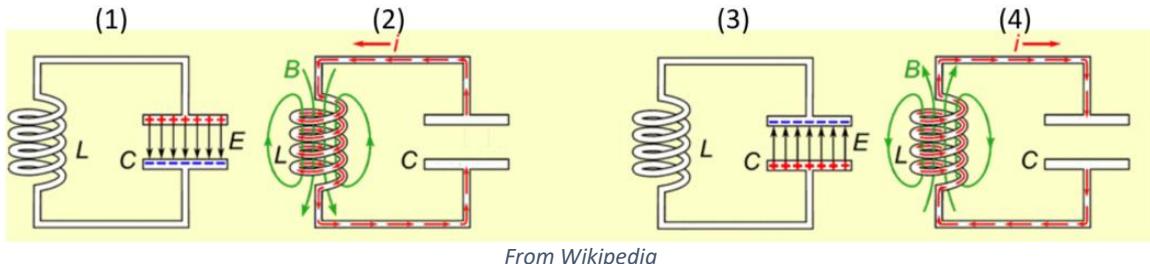
without inverters is linear. Moreover, the expression we derived in the beginning, for a circuit like this, also tells us the circuit is linear $V_o = \frac{Z_L}{Z_L + \frac{1}{N} \left(\frac{1}{CS} + R \right)} L\{v_{in} \tilde{n}\}$.

On the other hand, the case with inverters isn't linear. From simple calculations, we get that the relation $|V_{out}| = n * 64.52m$ is an approximation. Additionally, the output phase depends on how many cells are on. Thus, we conclude that the use of inverters causes the circuit to be non-linear. It makes sense, because inverters are made of transistors, which are non-linear components.

Predistortion can be applied to deal with this issue. In predistortion, the PA's inverse transfer function is applied to the input $f(f^{-1}(x)) = x$. As a result, a linear and nondistorted response is achieved by the system. This should cause the IQ constellation diagram to look more ideal.

Resonance

Natural frequency refers to the inherent frequency at which a physical system vibrates or oscillates when it is disturbed from balanced position and then left to move freely without any external forces acting on it. For example, an ideal LC circuit:



From Wikipedia

Initially, the capacitor is charged and has energy stored in the form of electric field (1). The capacitor's voltage drives current into the inductor and its electric field is transformed to magnetic field in the inductor. At some point, the voltage across the capacitor falls to zero (2) and the inductor's magnetic field induces voltage that causes current to recharge the capacitor, with opposite polarity to its original charge. The charging of the capacitor comes at a cost of the inductor's magnetic field. When the magnetic field is completely dissipated, the current will stop and the charge will again be stored in the capacitor, with the opposite polarity as before (3). Then the cycle will begin again, with the current flowing in the opposite direction through the inductor (4). It can be derived from a simple 2nd order differential equation that the voltage and current in the system oscillates at a frequency of $\omega_o = \frac{1}{\sqrt{LC}}$, this is the natural frequency.

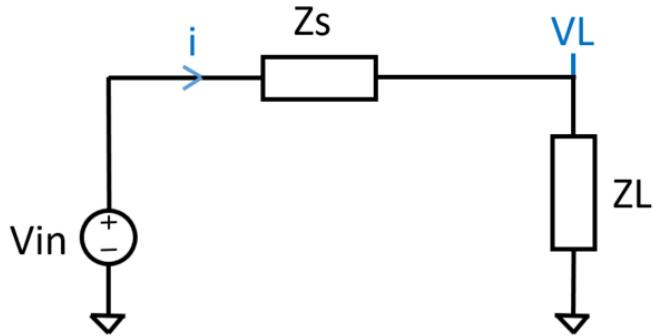
Resonance is a fundamental phenomenon that occurs in various physical systems when an external force or an applied signal matches the natural frequency of the system. When resonance occurs, the system responds with a significantly amplified response, resulting in increased energy transfer. In electronic circuits, resonance refers to the condition where the circuit "shows" maximum response (maximum power), when connected to an alternating input signal, which oscillates at the natural frequency of the system. When a circuit is in resonance, the load's reactive element is equals to minus the source's reactive element, namely $X_L = -X_s$.

Matching Network

We remind that Root Mean Square (RMS) is a mathematical concept used to describe the effective or "average" value of an alternating waveform. Denoting $f(t)$ as a periodic function with period T , the RMS is expressed as $f_{RMS} = \sqrt{\frac{1}{T} \int_{-T/2}^{T/2} f^2(t) dt}$. For sinusoidal signal $f(t) = A\sin(\frac{2\pi}{T}t)$ we get $f_{RMS} = \sqrt{\frac{1}{T} \int_{-T/2}^{T/2} A^2 \sin^2(\frac{2\pi}{T}t) dt} = \frac{A}{\sqrt{2}}$. For rectangular signal (0 to A) with duty cycle of 50% we get $\sqrt{\frac{1}{T} \int_0^{T/2} A^2 dt} = \frac{A}{\sqrt{2}}$, as well.

In our design, we are interested in maximum output power. The circuit's load is a 50Ω resistor, representing an antenna. We build a matching network between the IQ channels and the antenna. Its goal is to show the IQ channels an impedance, for which maximum power would be delivered to the antenna. The matching network is made of non-ideal transformers, which are also responsible for combining the I and Q components.

Design model



We model our design with input signal V_{in} , IQ arrays impedance Z_s and an impedance seen into the matching network Z_L , which includes both the matching network and antenna components and is the design model's load. We denote $Z_s = R_s + jX_s$ and we want to find $Z_L = R_L + jX_L$, such that we would get maximum power at the output.

The average power dissipated by the load Z_L is $P_L = \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} i^2(t) R_L dt = I_{RMS}^2 R_L = \{ \text{for sinusoidal or rectangular signal} \} = \frac{1}{2} |I|^2 R_L$. From Ohm's Law we know that $I = \frac{V_{in}}{Z_s + Z_L}$ so $|I|^2 = \left(\frac{|V_{in}|}{|Z_s + Z_L|} \right)^2 = \frac{|V_{in}|^2}{(R_s + R_L)^2 + (X_s + X_L)^2}$ and $P_L = \frac{1}{2} \frac{|V_{in}|^2 R_L}{(R_s + R_L)^2 + (X_s + X_L)^2}$. We want to maximize this expression and we can do that by minimizing the denominator. X_s and X_L are reactance elements, so their values can be both negative or positive. Thus, we can choose $X_L = -X_s$ to minimize the denominator. For example, if X_s is some capacitance element $X_s = -\frac{1}{\omega C}$ we can take an inductor $X_L = \omega L$ such that

for a specific frequency $\omega = \omega_o$, if we take $L = \frac{1}{\omega_o^2 C}$ we would get $X_s + X_L = 0$. ω_o is the resonant frequency of the circuit. $X_L = -X_s$ results in $P_L = \frac{1}{2} \frac{|V_{in}|^2 R_L}{(R_s + R_L)^2}$. Next, we want to find R_L that maximize the power. $\frac{d}{dR_L} \frac{R_L}{(R_s + R_L)^2} = \frac{(R_s + R_L)^2 - 2R_L(R_s + R_L)}{(R_s + R_L)^4} = \frac{R_s^2 - R_L^2}{(R_s + R_L)^4} = 0 \rightarrow R_s = R_L$, which can easily be proved as a maximum point. To summarize, in order to maximize the output power we want to pick a load of $Z_L = Z_s^*$.

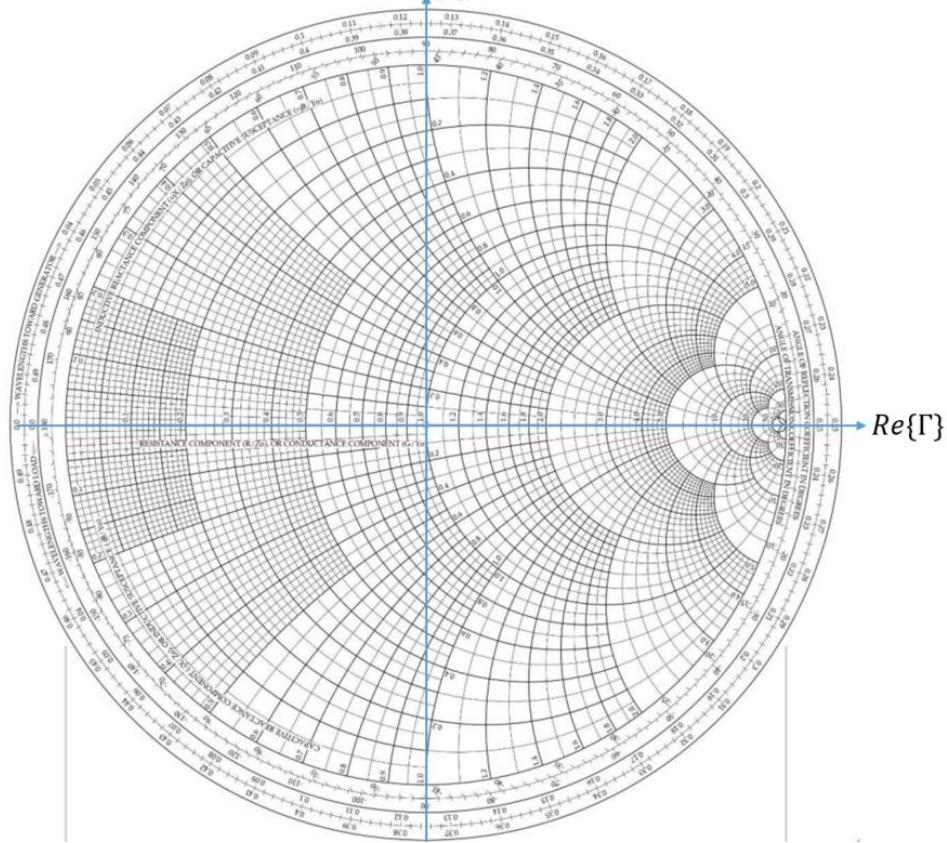
Therefore, it seems like we need to build the matching network such that the impedance seen into it is Z_L^* . However, our design specification is looser and doesn't require maximum output power. It requires, that the maximum output power, when all the cells are on, would be greater than $20dBm = 100mW$, at $2GHz$ frequency.

The Smith Chart

The Smith Chart is a graphical tool used in radio frequency (RF) and microwave engineering. Its goal is to assist in the analysis and design of transmission lines and impedance matching networks, which ensures maximum power transfer between source and load, or for the purpose of impedance transformation.

The Smith Chart

$$Im\{\Gamma\}$$



When source and load impedance are not complex conjugate of each other, maximum power transfer is not achieved. In this case power that was not transferred to the load would be reflected back to the source. The reflection coefficient is a measure of the reflection. It is the ratio between the voltage reflected back to the source and the total incident voltage $\Gamma = \frac{V_{reflected}}{V_{incident}} = \frac{Z_L - Z_0}{Z_L + Z_0}$. For no reflection $Z_L = Z_0$ and for maximum output power $Z_L = Z_0^*$. Here, Z_L is the impedance of the load and Z_0 is the characteristic impedance of the system, to which we want to match Z_L to. It must be noted that $|\Gamma| \leq 1$.

Usually, we use normalized impedance $z = \frac{z}{Z_0}$ (z is small z and Z is large z) for which we can write

$$(1) \Gamma = \frac{z_L - 1}{z_L + 1} = |\Gamma|e^{j\theta}, z_L = \frac{1 + |\Gamma|e^{j\theta}}{1 - |\Gamma|e^{j\theta}}$$

Moreover, we denote $z_L = r + jx, \Gamma = u + jv$ then from (1) we get

$$(2) z_L = r + jx = \frac{1 + u + jv}{1 - u - jv} = \frac{1 - u^2 - v^2}{(1 - u)^2 + v^2} + j \frac{2v}{(1 - u)^2 + v^2}$$

We can separate the real and imaginary parts and get

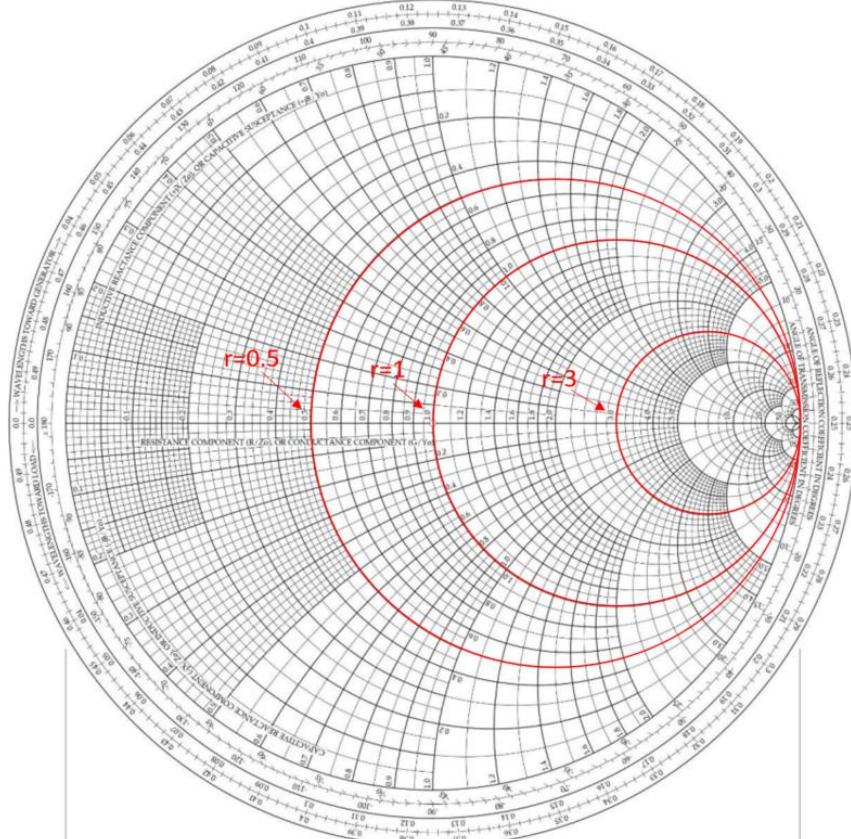
$$(3) \left(u - \frac{r}{1+r}\right)^2 + v^2 = \left(\frac{1}{1+r}\right)^2$$

$$(4) (u - 1)^2 + \left(v - \frac{1}{x}\right)^2 = \left(\frac{1}{x}\right)^2$$

The Smith Chart sits on a Cartesian coordinate system, where the Y axis is the imaginary value of the reflection coefficient $v = \text{Im}\{\Gamma\}$ and the X axis is the real value $u = \text{Re}\{\Gamma\}$. The center of the chart is the center of the Cartesian coordinate system. In this Cartesian coordinate equations (3) and (4) represent circles.

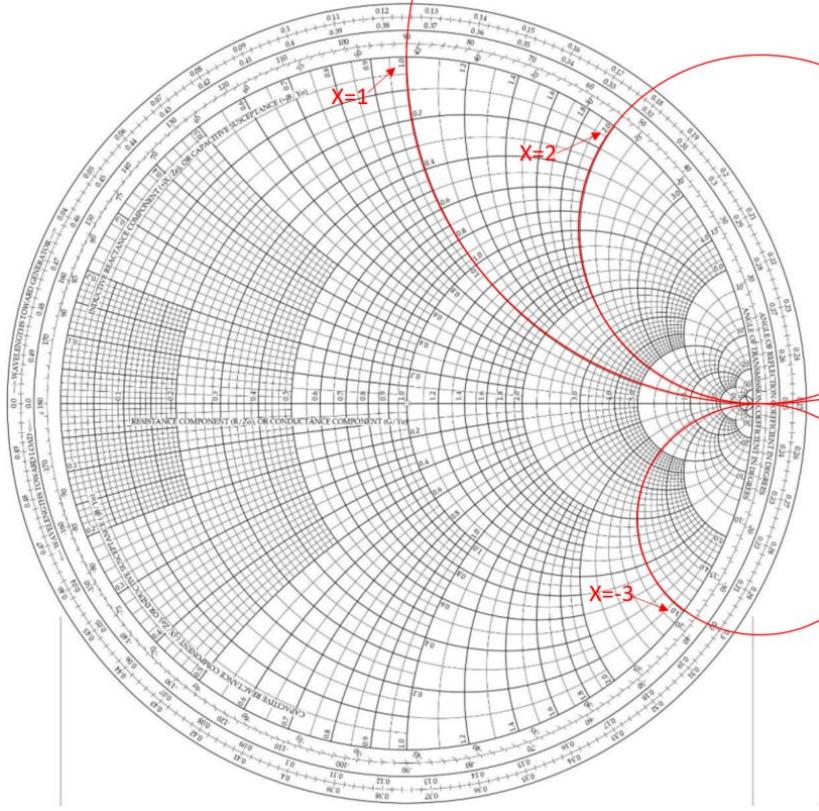
Equation (3) is a circle whose dimensions are determined by the resistive component of z_L (r). The circle's center point is at $(\frac{r}{1+r}, 0)$ (on the real axis) and it cuts the real axis at points $(1,0)$ and $(\frac{r-1}{r+1}, 0)$. When $r \rightarrow 0$ the circle's center is at $(0,0)$ and it cuts the real axis at points $(1,0)$ and $(-1,0)$. When $r \rightarrow \infty$ the circle is really a point at $(1,0)$. The Smith Chart plots equation (3) circles for different r values. Moreover, for some circles, it shows their respective r value near the point they cut the Real axis.

Examples of Equation (3) circles



Equation (4) is a circle whose dimensions are determined by the reactive component of z_L (x). The circle's center point is at $(1, \frac{1}{x})$ and it cuts the real axis only at point $(1,0)$, meaning that the circle sits on the Real axis. When $x \rightarrow 0$ the circle's center is at $(1, \infty)$ with radius ∞ , so it would look as if the Real axis is part of the circle. When $x \rightarrow \infty$ the circle is really a point at $(1,0)$. The Smith Chart plots equation (4) circles for different x values. Moreover, for some circles, it shows their respective x value near the point they cut the unit circle. For positive reactive components (inductors), the circle would be above the Real axis and for negative reactive components (capacitors), it would be under.

Example of Equation (4) circles



So, if we want to find the reflection coefficient for which $z_L = r_L + jx_L$ (the normalized impedance), we find the crossing point between the circle from equation (3) with $r = r_L$ and the circle from equation (4) with $x = x_L$. This point is the reflection coefficient.

The Smith Chart can also be used to plot normalized admittances $y_L = \frac{1}{z_L} = \frac{1-\Gamma}{1+\Gamma}$. Note that $y_L = \frac{1+\Gamma e^{j\pi}}{1-\Gamma e^{j\pi}}$, meaning that by rotating the impedance Smith Chart 180deg you get the admittance chart. If on our Smith Chart we have a point representing an impedance, we could rotate that point 180deg to get its admittance value.

If we denote $y_L = g + jb$ (g is conductance and b is susceptance) we can get equations as before

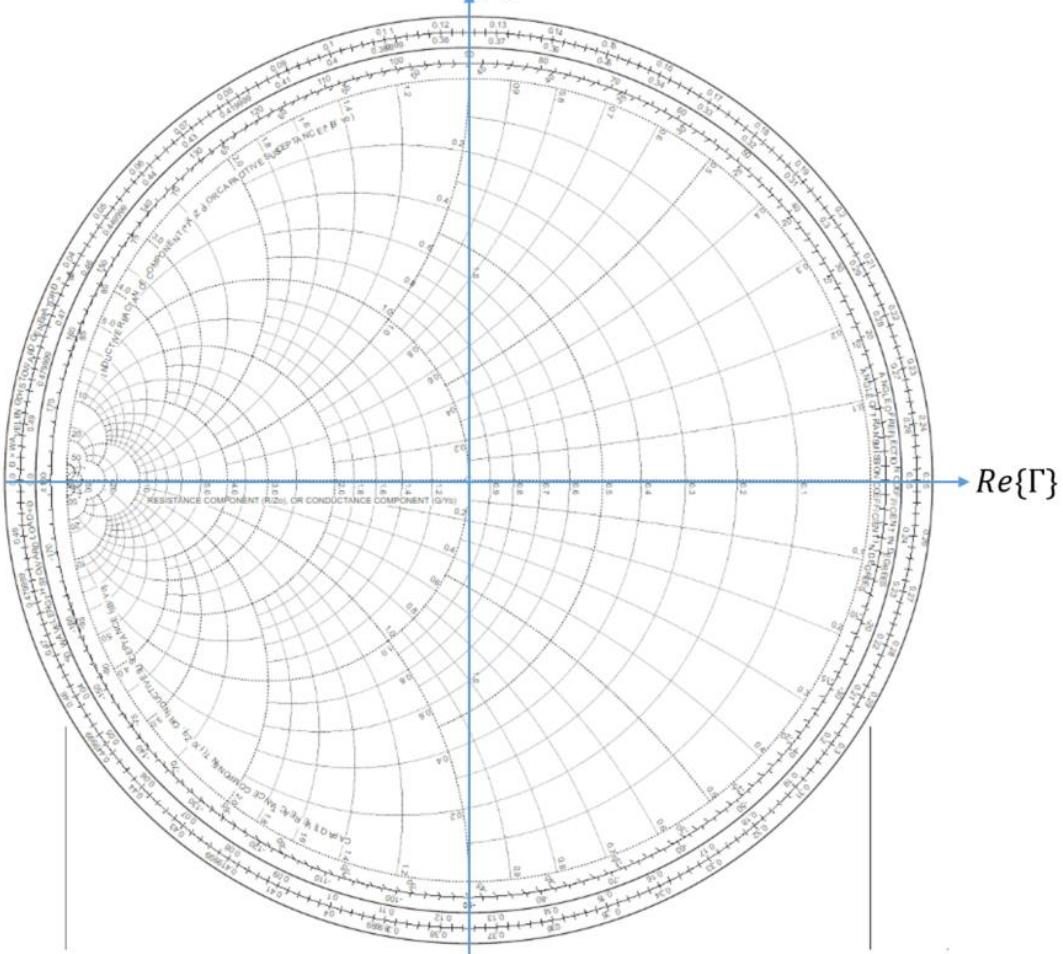
$$(5) \left(u + \frac{g}{1+g} \right)^2 + v^2 = \left(\frac{1}{1+g} \right)^2$$

Which is a circle whose origin at $(-\frac{g}{1+g}, 0)$ and cut the Real axis at points $(\frac{1-g}{1+g}, 0)$ and $(-1,0)$.

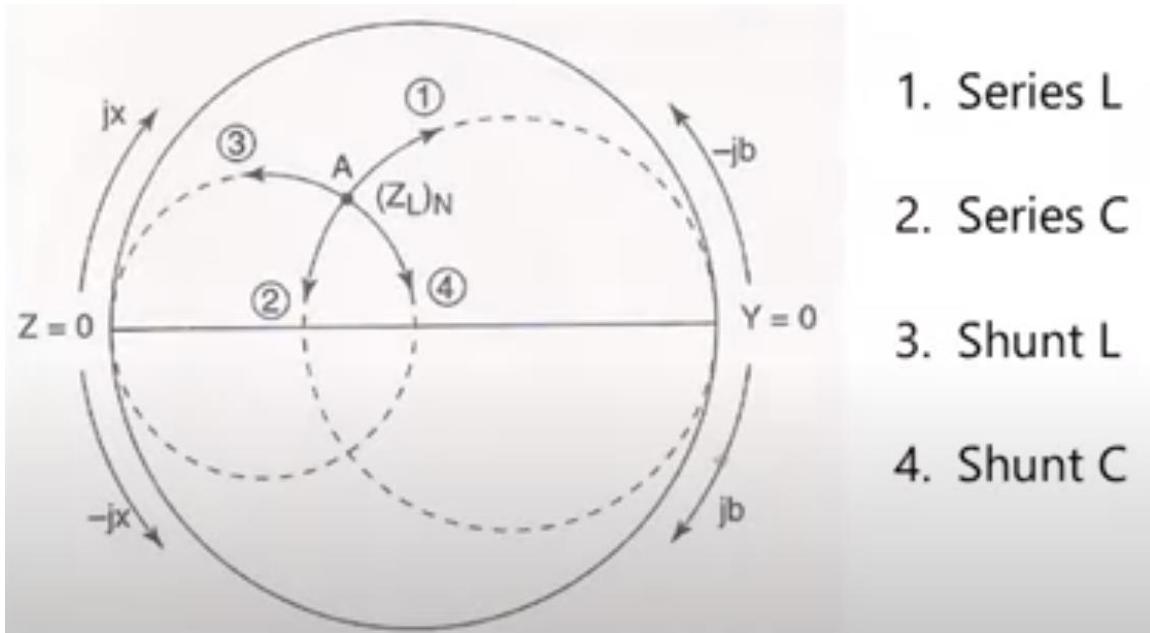
$$(6) (u+1)^2 + \left(v + \frac{1}{b} \right)^2 = \left(\frac{1}{b} \right)^2$$

Which is a circle whose origin at $(-1, -\frac{1}{b})$ and cuts the Real axis only at $(-1,0)$. For positive admittance reactive components (capacitors), the circle would be below the Real axis and for negative admittance reactive components (inductors), the circle would be above the Real axis.

The Admittance Smith Chart
 $Im\{\Gamma\}$



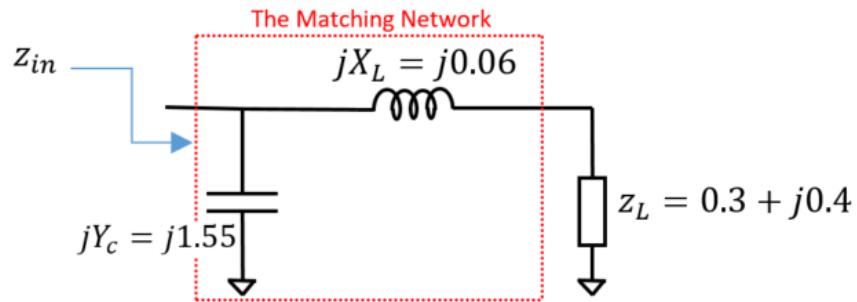
That means that given an initial normalized load z_L , if we connect a reactive element to the load the reflection coefficient will move as follows:



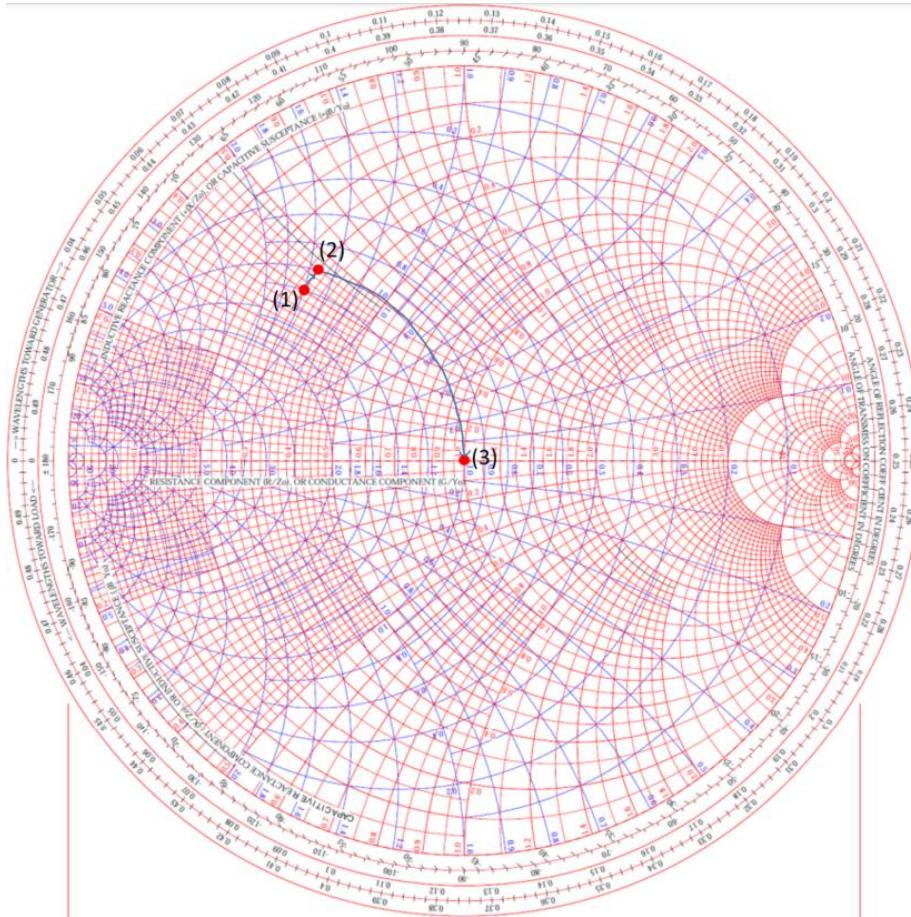
That is, when we connect a component in series, we move on the impedance circles and when we connect it in parallel, we move on the admittance circles. When adding a resistor in series, we stay on the imaginary impedance circle (equation 4), we were on, until we get to the right resistance. When adding a resistor in parallel, we stay on the imaginary admittance circle (equation 6), we were on, until we get to the right resistance.

For example, **for a certain frequency** let's say we have a load Z_L we want to match to our source impedance Z_s . Let's say that the normalized impedance equals to $z_L = \frac{Z_L}{Z_s} = 0.3 + j0.4$. We denote that point on the Smith Chart as (1). We want to build an efficient matching network, such that the reflection coefficient would be zero. Looking at the Smith chart (see figure below) we can do that by first moving the point (2) and then moving to point (3) (the center where $\Gamma = 0$). From point (1) to (2) we move on a constant resistance circle until we get to $x = 0.46$, thus we connect an inductor in series with impedance of $X_L = 0.06\Omega$. From point (2) to (3) we move on a constant conductance circle until we get to $b=0$, thus we connect a capacitor in parallel. At point (2) the susceptance is $b=-1.55$ (the admittance of an inductor is negative), so the admittance of the capacitor is $Y_c = 1.55\Omega$.

The Normalized Circuit



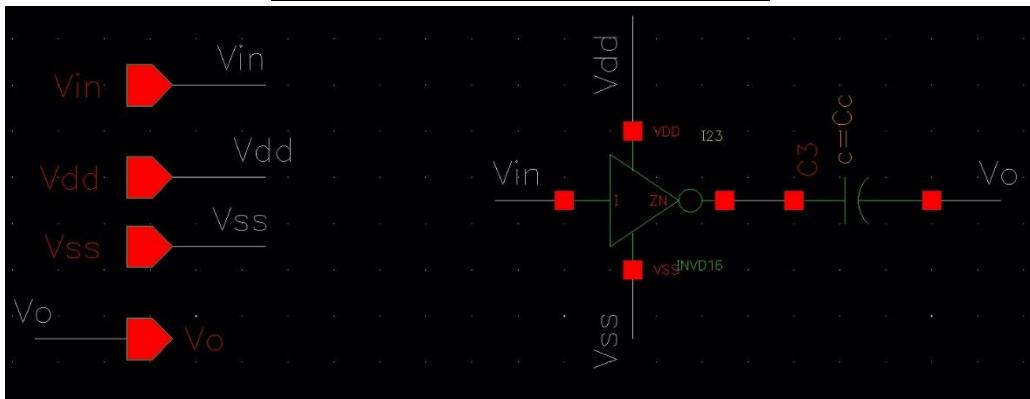
The Smith Chart used in the Example



Standard Cells

In our project we use standard, already made, logic gates. These gates come in fixed sizes. The transistors channel length is $60nm$, a PMOS finger width is $520nm$ and a NMOS finger width is $390nm$.

Branch Cell Made of Standard Inverter



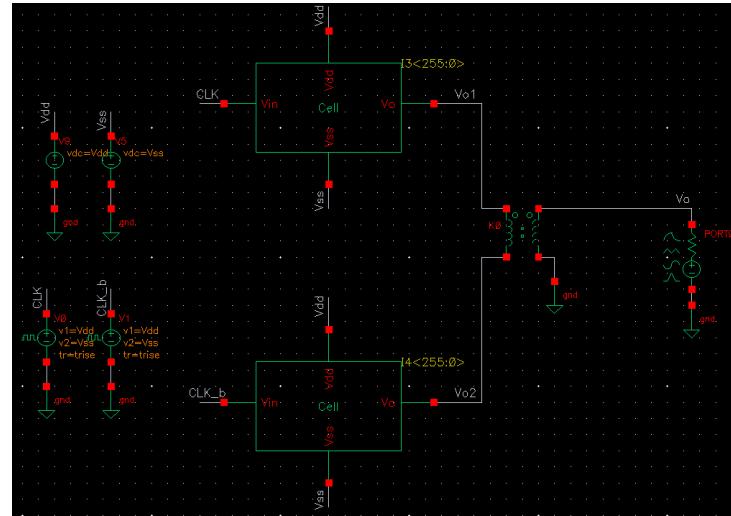
In our final design we use INVD16 as a cell inverter. Both its PMOS and NMOS transistors have 16 fingers. The capacitor is $C_c = 50fF$.

Load Pull

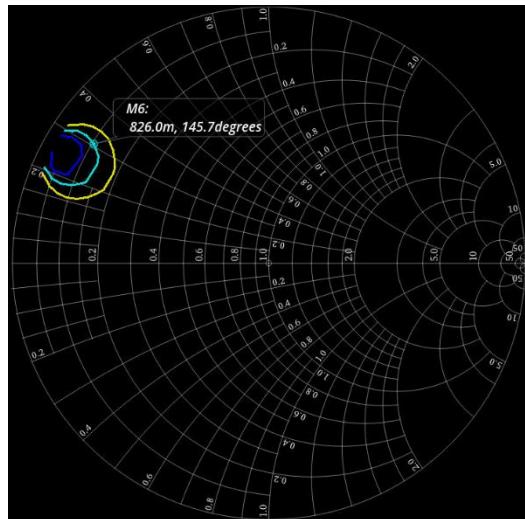
Because of the complexity of our circuit, it is difficult to determine the matching network parameters analytically. Thus, we use numerical simulation tool, Load Pull. Load Pull shows the output power for different reflection coefficients, at a specify frequency, which we can translate to the matching network parameters.

We begin by doing load pull on a single 256 cells channel. Load pull is done via harmonic balance simulation, when all the cells are on. For a more genuine simulation we set the rise and fall time of the clocks to 20ps.

Harmonic balance test bench



Load pull results – impedance smith chart

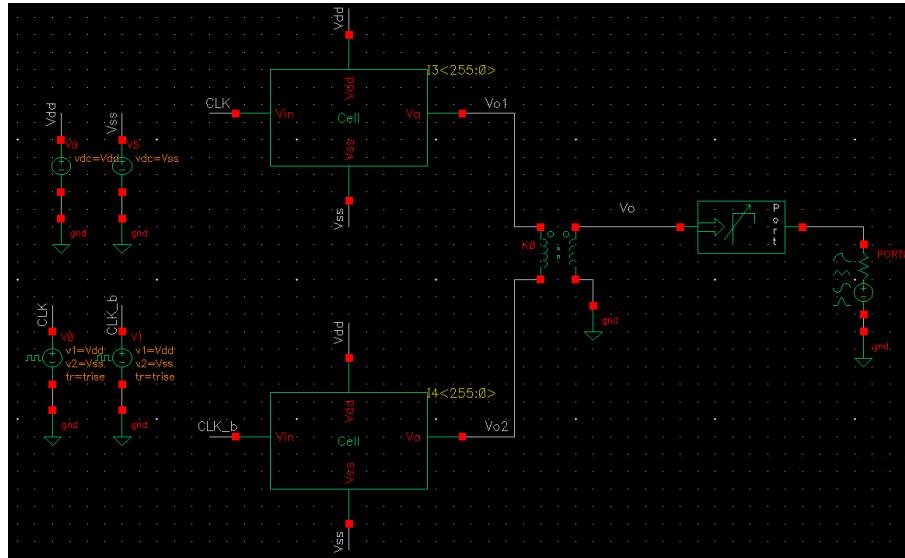


Each contour shows the possible reflection coefficients Γ , for which the output power of the fundamental frequency is 20dBm, 21dBm and 22dBm for the yellow, light blue

and blue contours, respectively. For example, for reflection coefficient of $\Gamma = 0.826\angle 145.7^\circ$, the output power of the fundamental frequency should be $21dBm$.

We can verify this example by using “portAdaptor” component from “rfExamples” library. It lets us set the reflection coefficient we want our circuit to see. We set it to $0.826\angle 145.7^\circ$.

Reflection coefficient test bench

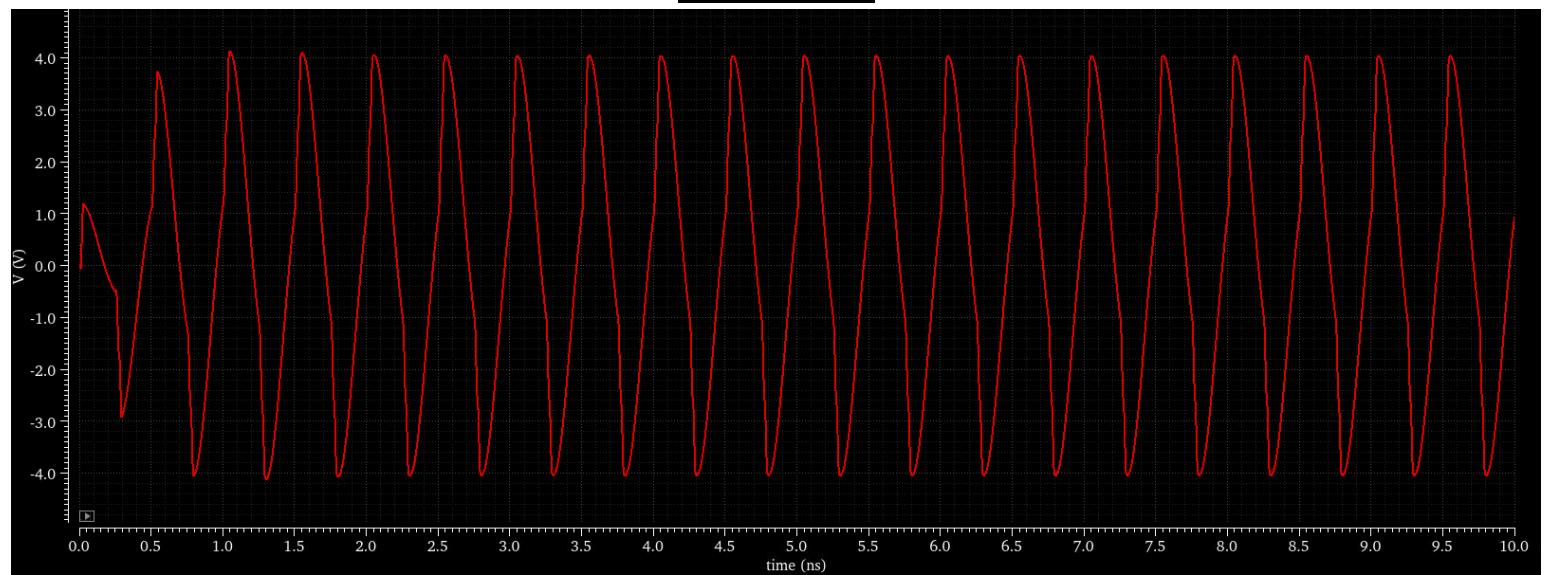


Expressions used

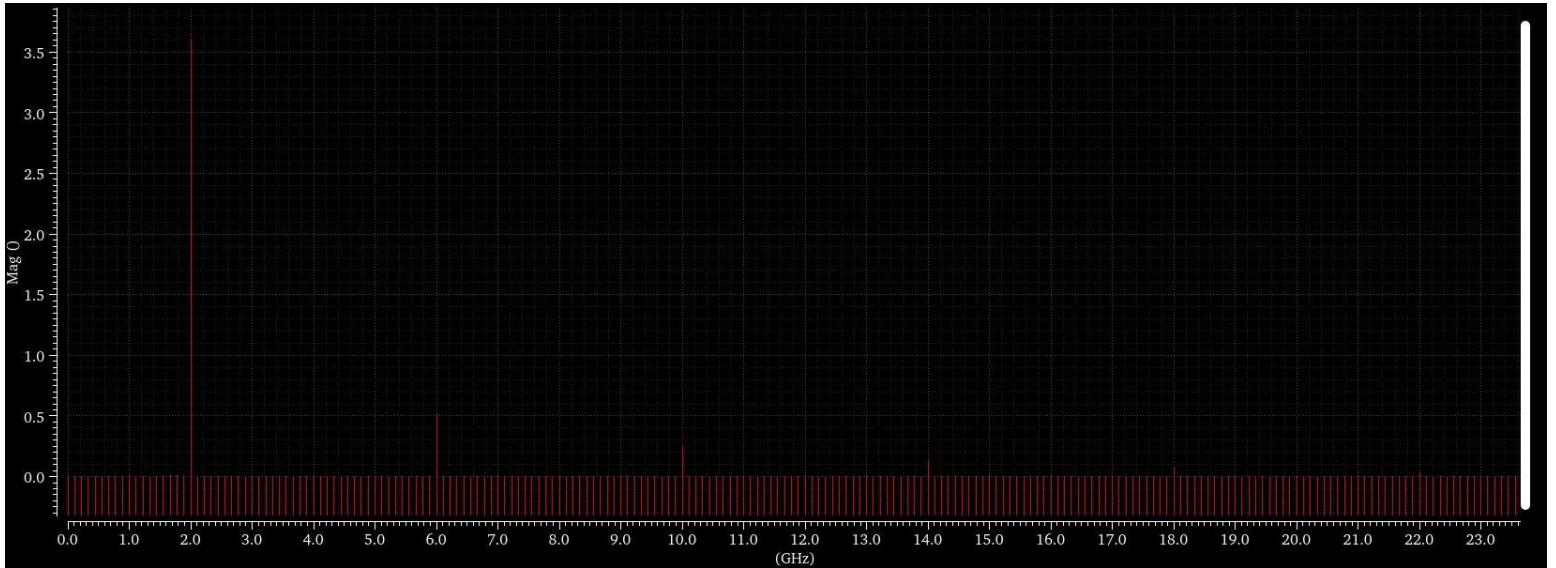
Vout	expr	vtime("tran "/Vo")
Iout	expr	itime("tran "/PORT0/PLUS")
Pout_h1 [dBm]	expr	dBm((0.5 * real(value((dft(Vout VAR("t1") VAR("t2") (2**14) "Rectangular" 1 1 1.0) * conjugate(dft(Iout VAR("t1") VAR("t2") (2**14) "Rectangular" 1 1 1.0)) VAR("frf")))))
Vout spectrum	expr	dft(Vout VAR("t1") VAR("t2") (2**14) "Rectangular" 1 1 1.0)

Using the following expressions we extract the output voltage, the output power of the first harmonic and the output voltage spectrum. We get that the output power of the first harmonic is $21.17dBm$, which is very close to $21dBm$. Load pull finds the output power for finite number of reflection coefficient and gives approximated contours. Thus, the reflection coefficient $0.826\angle 145.7^\circ$ is an approximated value for $21dBm$ and hence the error.

Output voltage



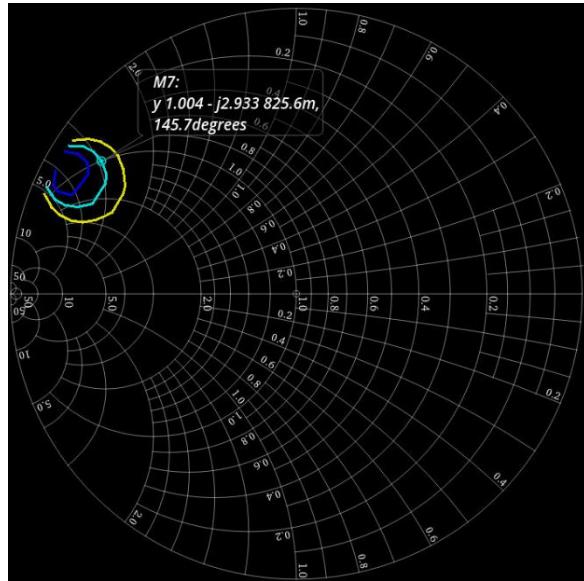
Output voltage – spectrum



The output voltage signal is fairly sinusoidal, meaning good filtering. Additionally, by looking at the spectrum of the signal, we see that the non-fundamental harmonics are relatively small, indicating good filtering.

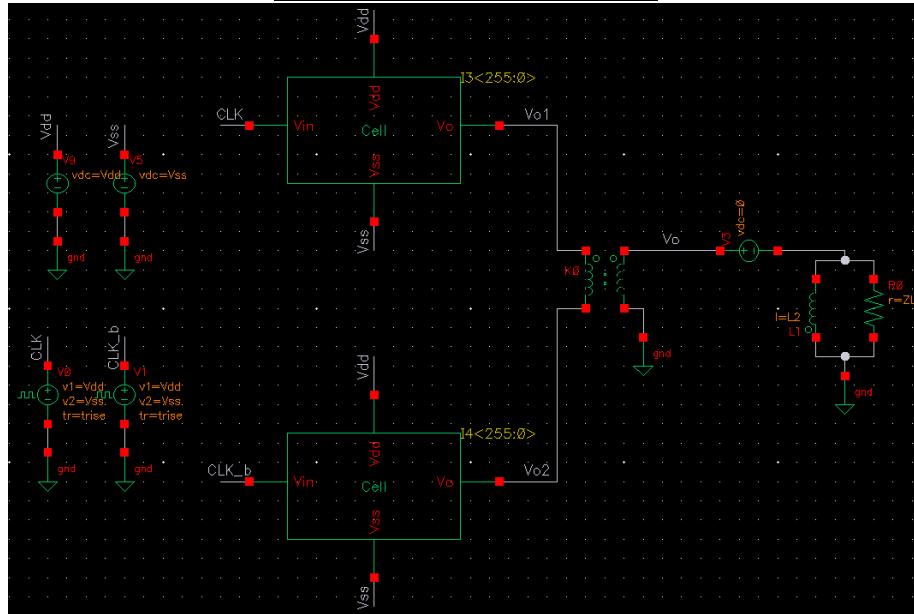
By looking at the load pull results on the admittance smith chart, we can easily derive a matching network that will result in $21dBm$ output power.

Load pull results – admittance smith chart



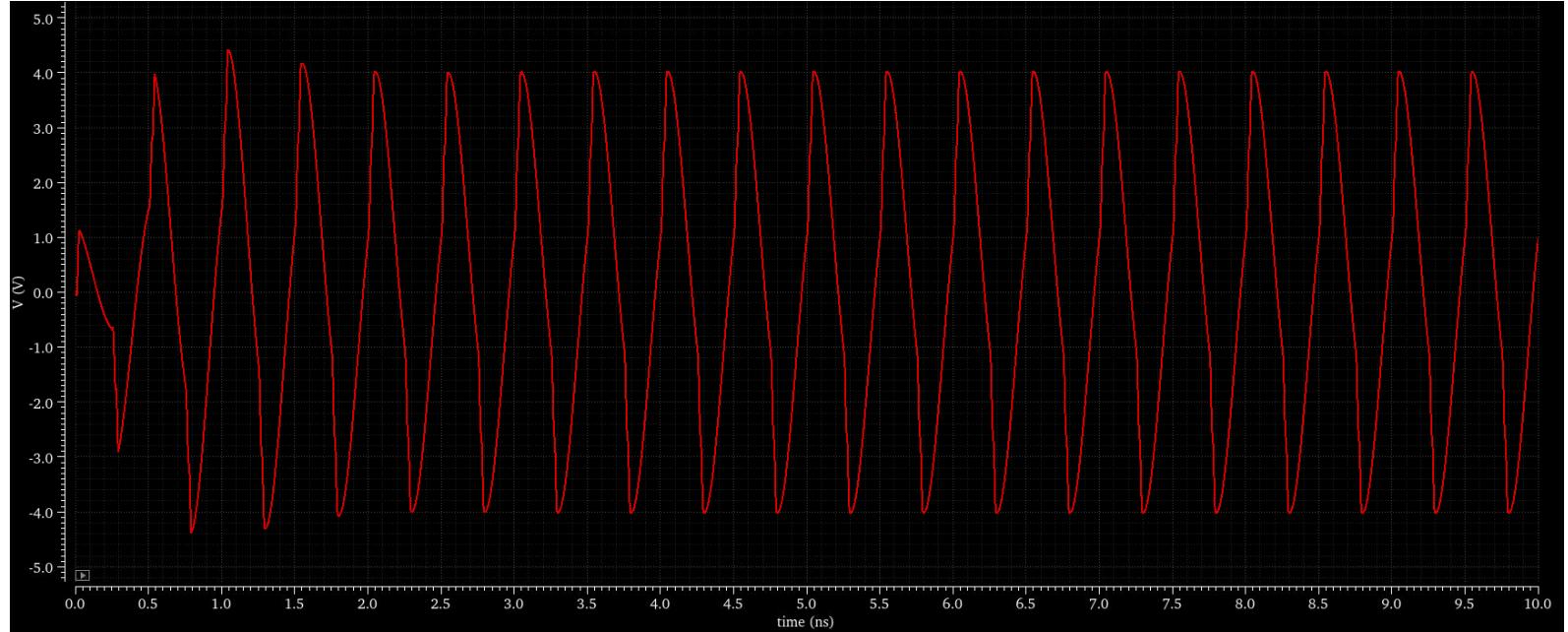
We can get to the $21dBm$ contour only by moving on the conductance circle. We move on the conductance circle counterclockwise by connecting an inductor in parallel. For output power of $21dBm$ the inductor's admittance needs to be 2.933 . Thus, for frequency of $2GHz$ the inductor normalized inductance is $\tilde{L} = \frac{1}{2\pi f * 2.933} = 27.13pH$. The real value of the inductance is $L = 50 \cdot \tilde{L} = 1.357nH$, because the matching is done for a 50Ω load.

Matching network test bench



The load is a 50Ω resistor, representing the antenna. The matching network is made only of an inductor $L = 1.357nH$ in parallel to the load. From simulation, we get that the output power of the fundamental harmonic is $21.2dBm$.

Output voltage using the matching network

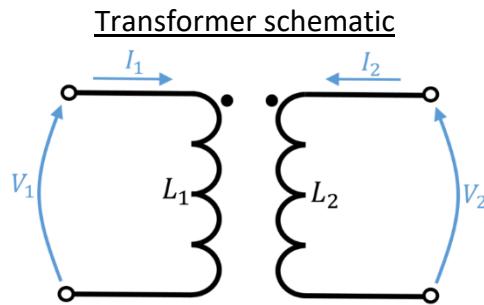


We see that the output voltage signal is similar to the one we got for the reflection coefficient simulation.

Non-Ideal Transformers

In our project, transformers are used for adding the I and Q components, moving from differential to single ended output and for matching the load. The transformers are not ideal and we must count for losses and high frequency effects.

Transformers are made of two inductors, input and output, in parallel. The input inductor is called the primary winding, and the output inductor is called the secondary winding. When an alternating current passes through the primary winding, it creates a changing magnetic field. This changing magnetic field induces a voltage in the secondary winding, which is then transferred to the load.



We denote the inductance, current and voltage of the primary winding as L_1 , I_1 and V_1 , respectively. Similarly, for the primary winding we denote L_2 , I_2 and V_2 .

We define k as the coupling coefficient of the inductors. It is a measure of the coupling between the primary and secondary windings of a transformer. It describes the extent to which the magnetic field generated by the current in the primary winding couples with the turns of the secondary winding. The coupling coefficient is a dimensionless quantity and its value ranges from zero to one.

Ideal transformer has $k = 1$, indicating that all the magnetic flux generated by the primary winding links with the turns of the secondary winding. This results in efficient energy transfer from the primary to the secondary. In practice, some degree of leakage flux and imperfections are present, so $0 < k < 1$. A typical value for k is 0.8.

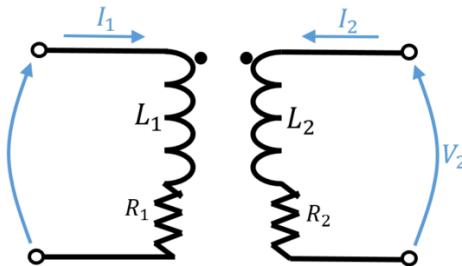
The transformer behavior is represented by two equations $V_1 = L_1 \frac{dI_1}{dt} + k\sqrt{L_1 L_2} \frac{dI_2}{dt}$ and $V_2 = L_2 \frac{dI_2}{dt} + k\sqrt{L_1 L_2} \frac{dI_1}{dt}$. Ideally, when $k = 1$ we get $\frac{V_2}{V_1} = \sqrt{\frac{L_2}{L_1}}$.

An important characteristic of non-ideal inductors, mainly at high frequencies, is the quality factor (Q factor). It is a measure of their efficiency in storing and releasing energy. It is defined as $Q = \frac{\omega L}{R}$, where ωL is the inductor's reactance and R is the inductor's effective resistance. Transformer with high Q factor inductors, means minimal energy lost by the transformer, and maximal power transfer to the load. A typical value for Q at high frequencies is between 10 – 15.

At any frequency, an inductor's winding has some resistance that causes dissipation of energy. At high frequencies, inductor's electric current tends to mainly flow near its surface, a phenomenon called the Skin effect. It causes the resistance of the inductor to increase and its Q factor to decrease. Additionally, at high frequencies, parallel inductors close to each other suffer from the Proximity effect. It causes the inductors' current to be concentrated in a thin strip on the side near the adjacent inductors. Like the Skin effect, it causes the inductors' resistance to increase and their Q factor to decrease.

Another important phenomenon of parallel inductors is parasitic capacitance. At high frequencies, the capacitance between the inductors becomes more influential. It is not a cause of energy loss, but can change the behavior of the inductors.

Transformer schematic – finite Q factor

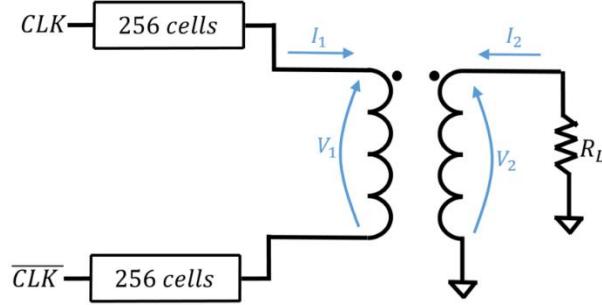


At high frequencies, when the Q factor is not finite we must take into account the effective resistance of the inductors. We model the resistance in series with the inductors. The primary winding resistance is $R_1 = \frac{\omega_o L_1}{Q_1}$ and the secondary resistance is $R_2 = \frac{\omega_o L_2}{Q_2}$. The transformer equations are now $V_1 = L_1 \frac{dI_1}{dt} + \frac{\omega_o L_1}{Q_1} I_1 + k\sqrt{L_1 L_2} \frac{dI_2}{dt}$ and $V_2 = L_2 \frac{dI_2}{dt} + \frac{\omega_o L_2}{Q_2} I_2 + k\sqrt{L_1 L_2} \frac{dI_1}{dt}$.

Matching using Non-Ideal Transformer

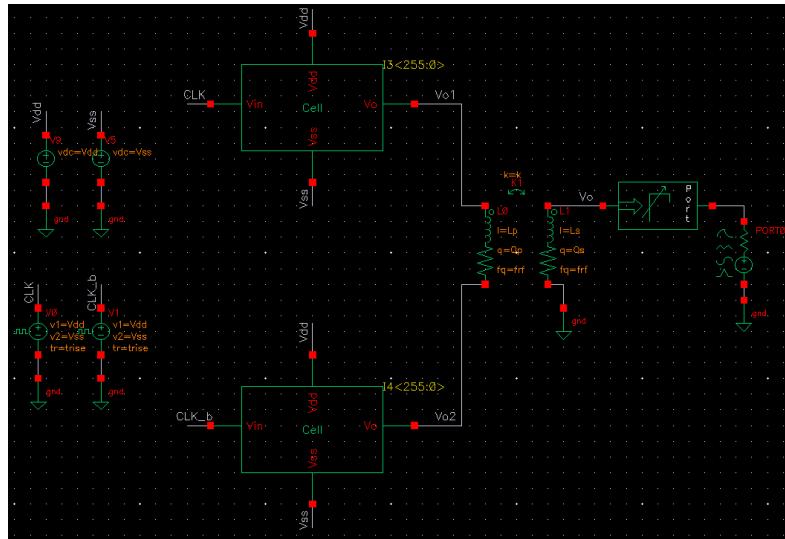
We want to match our load using the non-ideal transformer. First of all, we look at the simplest case where $k = 1$ and $Q \rightarrow \infty$. From the load pull simulation, we saw that our matching network is an inductor in parallel to the load. We want to prove that we get that when using non-ideal transformer.

Matching using non-ideal transformer – schematic



We denote $\alpha = \sqrt{\frac{L_2}{L_1}}$. For $k = 1$ and $Q \rightarrow \infty$, in Laplace domain we have $V_1 = sL_1(I_1 + \sqrt{\alpha}I_2)$ and $V_2 = s\sqrt{\alpha}L_1(I_1 + \sqrt{\alpha}I_2)$. Using the fact that $V_2 = -I_2R_L$, we get $I_2 = -\frac{s\sqrt{\alpha}L_1}{R_L+s\alpha L_1}I_1$, so $V_1 = sL_1I_1\frac{R_L}{R_L+s\alpha L_1}$. Thus, the impedance seen by the cells is $\frac{V_1}{I_1} = \frac{R_LsL_1}{R_L+s\alpha L_1} = \frac{1}{\alpha}(R_L||s\alpha L_1)$, which is an inductor in parallel to the load, just as we wanted. If the inductances are equal, $\alpha = 1$ and the impedance is $R_L||sL_1$.

Matching network test bench



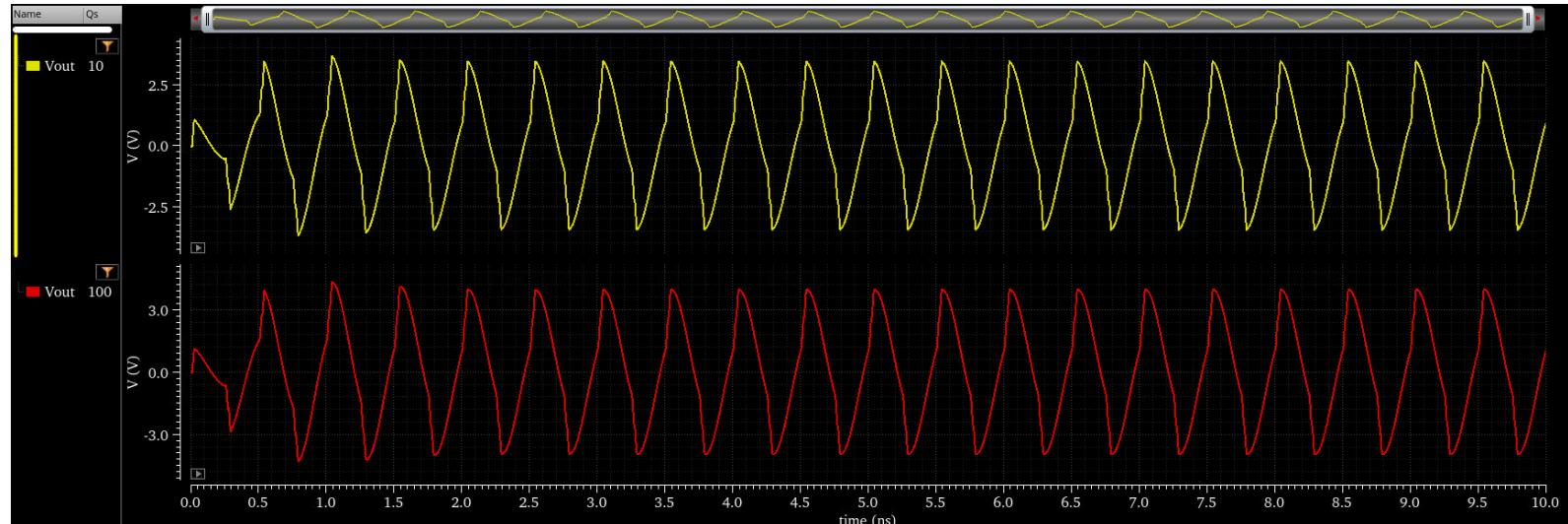
We use the port adaptor only for simulation reasons. Its reflection coefficient is set to zero.

We begin by simulating the case where $k = 1$, $\alpha = 1$ and Q varies and the same for both inductors. From the load pull simulation, we saw that an inductor of $1.357nH$, in parallel to the load, is sufficient to get $21dBm$ output power. Thus, we pick $L_1 = L_2 = 1.357nH$.

Simulation output power result for different Q factors

Qp Filter	Qs Filter	Pout_h1 Filter
10	10	19.22
20	20	20.18
30	30	20.51
40	40	20.68
50	50	20.79
60	60	20.86
70	70	20.91
80	80	20.94
90	90	20.97
100	100	21

Simulation output voltage for Q factor 10 and 100



For low Q values, more power is dissipated on the transformer's inductors and less power is transferred to the load. As the Q factor increases the output power does as well. However, it is saturated to about $21dBm$, as we got in the ideal transistor simulation, at the load pull section.

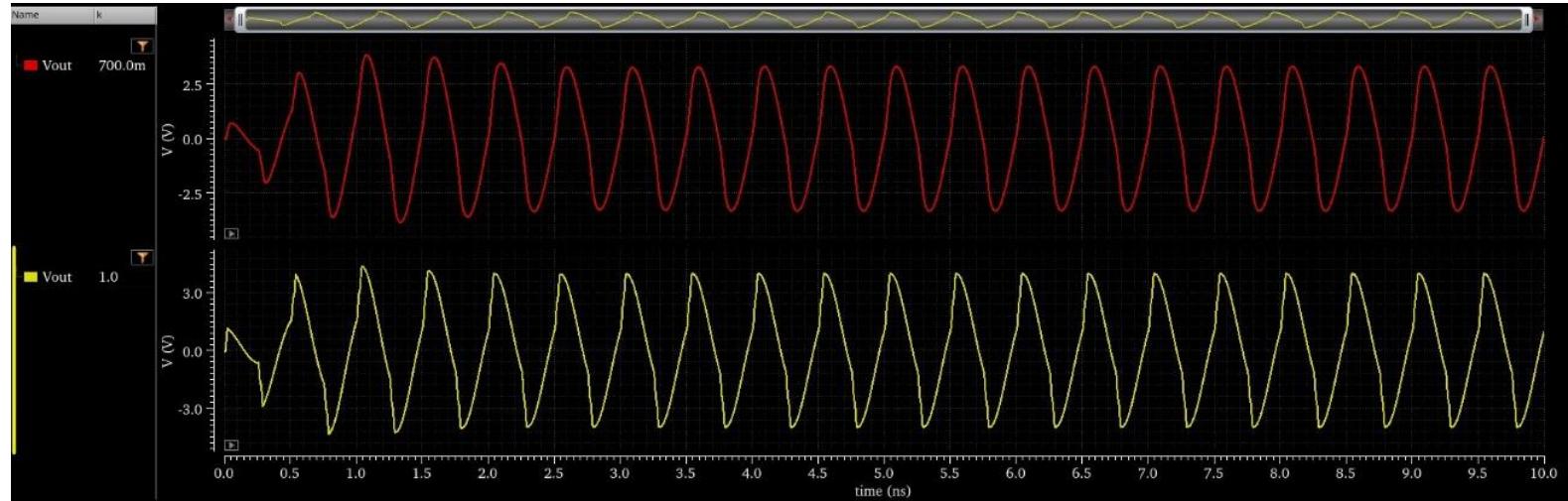
The yellow waveform is the output voltage for $Q = 10$ and the red waveform is for $Q = 100$. The red waveform is similar to the one we got in the ideal transistor case.

Next, we simulate the case where $\alpha = 1$, $Q = 200$ and k varies. Inductors with such a large Q factor can be considered ideal. Again, we pick $L_1 = L_2 = 1.357nH$.

Simulation output power result for different k factors

k	Pout_h1
Filter	Filter
700m	20.25
720m	20.39
740m	20.52
760m	20.62
780m	20.72
800m	20.8
820m	20.87
840m	20.93
860m	20.98
880m	21.01
900m	21.03
920m	21.05
940m	21.05
960m	21.05
980m	21.03
1	21.01

Simulation output voltage for k factor 0.7 and 1



For low k values, there is more flux leakage from the primary to the secondary, which results in smaller energy transfer between the inductors and less power transferred to the load. As k increases the output power does as well. However, it is saturated to about $21dBm$, as we got in the ideal transistor simulation, at the load pull section. We see that the maximum output power is for $k < 1$, but we believe this is a simulation inaccuracy.

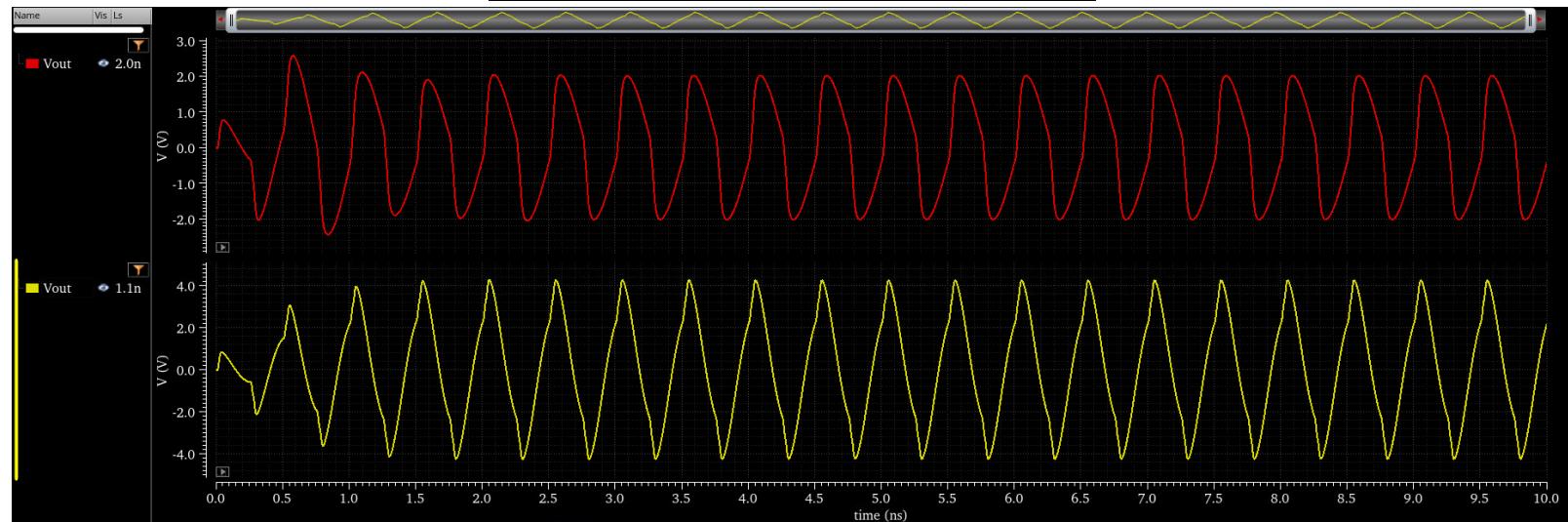
The red waveform is the output voltage for $k = 0.7$ and the yellow waveform is for $k = 1$. The yellow waveform is similar to the one we got in the ideal transistor case, but the red one seems more sinusoidal.

Finally, we simulate a fairly realistic case where $k = 0.8$, $Q = 14$ and the inductances vary. We keep $L_1 = L_2$, for simplicity.

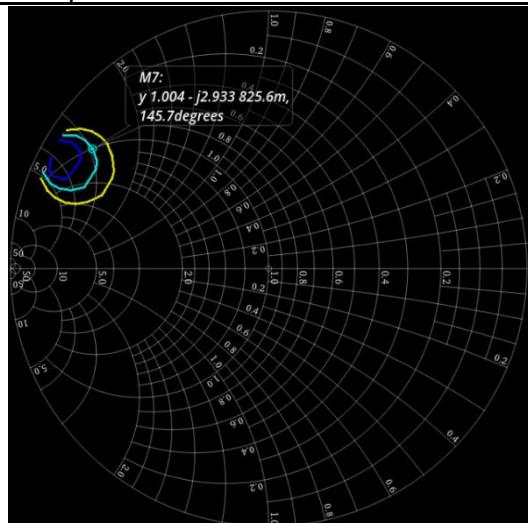
Simulation output power result for different L values

Lp	Ls	Pout_h1
Filter	Filter	Filter
2n	2n	16.06
1.9n	1.9n	16.43
1.8n	1.8n	16.83
1.7n	1.7n	17.29
1.6n	1.6n	17.8
1.5n	1.5n	18.4
1.4n	1.4n	19.07
1.3n	1.3n	19.82
1.2n	1.2n	20.6
1.1n	1.1n	21.24
1n	1n	21.44

Simulation output voltage for L 2nH and 1.1nH



Load pull results – admittance smith chart



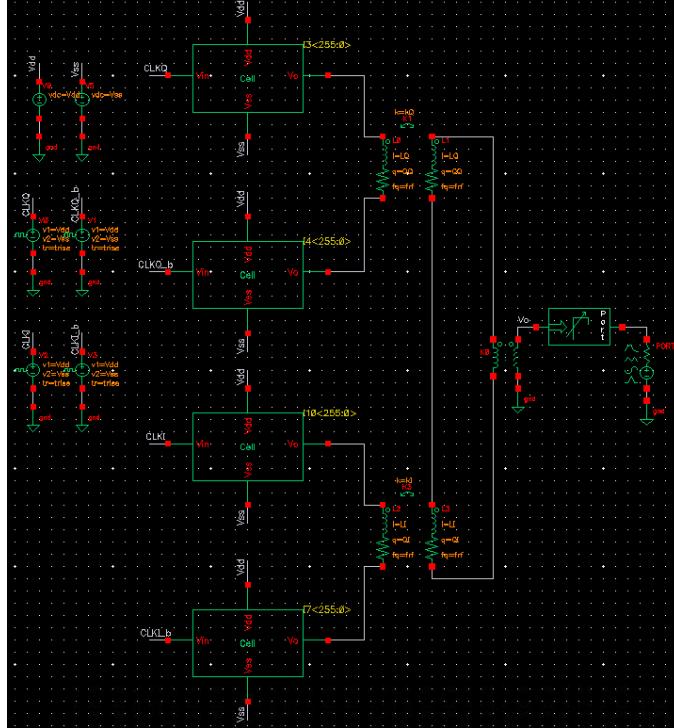
For $k = 0.8$, $Q = 14$ the transformer's power loss is high. In this case, for the inductance taken before, the output power is below 20dBm . As the inductance decreases the output power increases. It makes sense when we look at the admittance smith chart, from the load pull section. As L decreases, $\frac{1}{\omega_0 L}$ increases and we move on the conductance circle counterclockwise. As a result, we get into the more inner power contours, which correspond to larger output powers. If we keep decreasing L , at some point we would start moving away from the inner contours and the output power would decrease.

The red waveform is the output voltage for $L = 2nH$ and the yellow waveform is for $L = 1.1nH$. The yellow waveform is more similar to the one we got before for the 21dBm case.

Matching using Non-Ideal Transformer – Two channels

We look at the case where both I and Q channels are connected. Assuming the primary winding is the same as the secondary, we denote the I and Q transformers' parameters k_I, L_I, Q_I and k_Q, L_Q, Q_Q , respectively. To move from differential output to single ended output we use an ideal transformer.

Two Channel Matching Network Test Bench

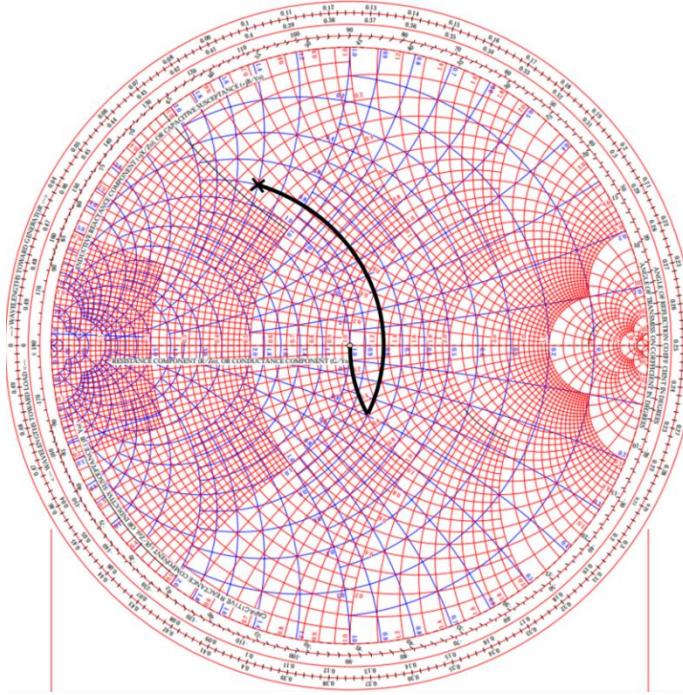


We begin by finding the effect of one channel on the other, in an ideal case where $k = 1$ and $Q \rightarrow \infty$. Without the loss of generality, we find how the I channel effects the Q channel. Approximately, we can model 256 cells in parallel as a capacitor $256C_c$. As we saw before, when we mirror an impedance from one winding to the other, we get the impedance seen by the one winding, in parallel to the transformer's inductance.

The I channel primary winding sees two $256C_c$ capacitors in series, which is equivalent to a $128C_c$ capacitor. By mirroring to the secondary we get $sL_I \parallel \frac{1}{s128C_c}$. The load is connected to an ideal transformer, so when we mirror it we just get the load R_L . As we mirror the impedance seen by the Q channel secondary winding, we get the impedance seen by the Q channel $Z_{Q,out} = sL_Q \parallel (R_L + sL_I \parallel \frac{1}{s128C_c})$.

From hand calculations, $sL_I \parallel \frac{1}{s128C_c}$ acts like a capacitor in the range of inductors we use $1 < L$. Thus, the Q channel sees a capacitor in series with the load.

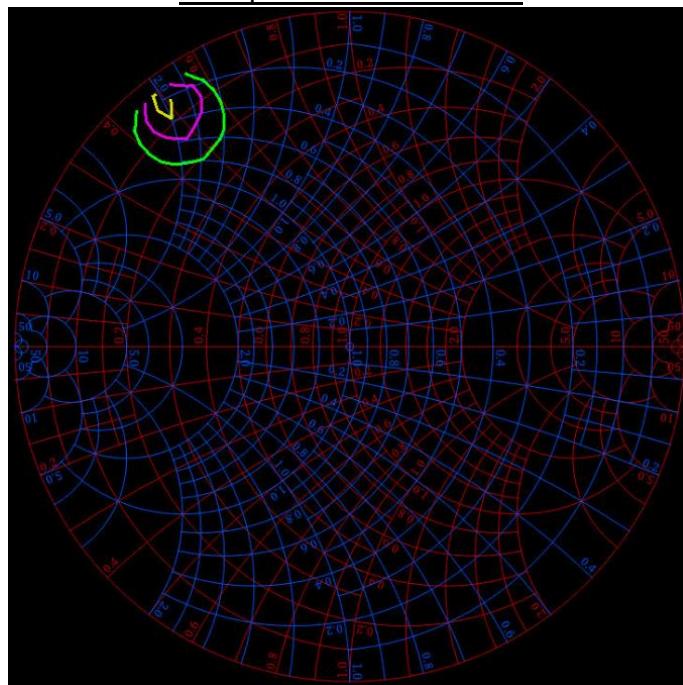
The effect of the series capacitor on the reflection coefficient



As a result, we expect the output power to be lower than the case of only one channel connected. We simulate this case, with $k = 1$, $Q \gg 10$ and $L_Q = L_I = 1.357nH$. For these parameters, the one channel simulation resulted in an output power of about $21dBm$. On the other hand, this simulation results in an output power of $17.54dBm$, which is much lower.

Using ideal transformers, we perform a load pull simulation for the two channel case, when only the Q channel is on.

Load pull simulation result



Each contour shows the possible reflection coefficients Γ , for which the output power of the fundamental frequency is $19dBm$, $20dBm$ and $21dBm$ for the green, purple and yellow contours, respectively. We see that an inductor in parallel to the load is not enough to even get to the $19dBm$ contour. To get there, we would need an inductor in series with the load.

For example, we can connect an inductor in parallel to the load with $Y_{L,p} = -3\Omega^{-1}$ (normalized). Then, we connect an inductor in series with $X_{L,s} = 0.2\Omega$ (normalized), to get about $21dBm$. The series inductor non-normalized reactance is 10Ω . The reactance seen into the I channel is $X_c = \frac{-1}{128C_c\omega_o}$. For $C_c = 50fF$ and $f_o = 2GHz$, $X_c = -12.43\Omega$.

This reactance is about the same as the series inductor's reactance, but with opposite sign. Thus, the series inductor reduces the effect of the I channel impedance. However, in our design, we don't use a series inductor and we can't get above $20dBm$ output power, when only one channel is on.

Ideally, when both channels are on, both contribute orthogonal components with the same amplitude, to the output. Thus, at the fundamental frequency, we expect the output voltage amplitude to be $\sqrt{2}$ times the input's amplitude. As a result, when both channels are on, we expect the output power to be $10 \log((\sqrt{2})^2) = 3.01dBm$ larger, than when only one channel is on. However, due to non-linearity effects, the output voltage and power we get are different.

We simulate the circuit both when one channel is on and when two channels are on. We begin with setting $k = 1$, $L_Q = L_I = 1.2nH$ and Q varies.

Simulation output power result for different Q factors

QQ	QI	One channel Vout_h1	One channel Pout_h1	Two channel Vout_h1	Two channel Pout_h1
Filter	Filter	Filter	Filter	Filter	Filter
10	10	2.067	16.31	2.908	19.27
20	20	2.268	17.11	3.108	19.85
30	30	2.343	17.4	3.165	20.01
40	40	2.382	17.54	3.191	20.08
50	50	2.406	17.63	3.205	20.12
60	60	2.422	17.68	3.215	20.14
70	70	2.434	17.72	3.221	20.16
80	80	2.442	17.76	3.225	20.17
90	90	2.449	17.78	3.229	20.18
100	100	2.455	17.8	3.232	20.19

For low Q values, more power is dissipated on the transformer's inductors and less power is transferred to the load. As the Q factor increases the output power does as well. However, it is saturated to about $17.8dBm$.

Next, we simulate the case where $Q \gg 10$, $L_Q = L_I = 1.2nH$ and k varies. Inductors with such a large Q factor can be considered ideal.

Simulation output power result for different k factors

kQ	kl	One channel Vout_h1	One channel Pout_h1	Two channel Vout_h1	Two channel Pout_h1
Filter	Filter	Filter	Filter	Filter	Filter
800m	800m	2.926	19.32	3.679	21.31
820m	820m	2.884	19.2	3.641	21.22
840m	840m	2.841	19.07	3.602	21.13
860m	860m	2.797	18.93	3.561	21.03
880m	880m	2.752	18.79	3.518	20.92
900m	900m	2.708	18.65	3.474	20.82
920m	920m	2.663	18.51	3.429	20.7
940m	940m	2.618	18.36	3.384	20.59
960m	960m	2.574	18.21	3.339	20.47
980m	980m	2.531	18.07	3.294	20.35
1	1	2.488	17.92	3.248	20.23

We would of thought that for low k values, there is more flux leakage from the primary to the secondary, which results in smaller energy transfer between the inductors and less power transferred to the load. However, this is not the case. The smaller the k the larger the output power, both when one and two channels are on. However, if we kept decreasing k , at some point we would of seen the output power drop.

To understand this result, we need to understand the effect of mirroring when $k \neq 1$. We assume the primary and secondary inductances are equal and a load Z_L is connected to one side. Then, mirroring of the load to the other side results in $sL||Z_L(1 + \frac{sL}{Z_L}(1 - k^2))$. When $k = 1$ we get $sL||Z_L$ as before.

Previously, we found out that the capacitive effect of one channel on the other, makes the matching worse and causes the output power to decrease. That is because the channels see a capacitance in series with the load.

The capacitance in series with the load is the mirrored impedance of a channel. The impedance of a channel is approximately $\frac{1}{j\omega 128C_c}$. Thus, the mirrored impedance is $Z_m = j\omega L||\frac{1}{j\omega 128C_c}(1 - \omega^2 128C_c L(1 - k^2))$. When $f_o = 2GHz$, $C_c = 50fF$ and $L = 1.2nH$, Z_m is a negative reactance, which is a capacitive impedance. As k decreases, Z_m decreases, the series capacitance effect decreases, so the output power increases, as the simulation shows.

Finally, we simulate a fairly realistic case where $k = 0.8$, $Q = 14$ and the inductances vary.

Simulation output power result for different L factors

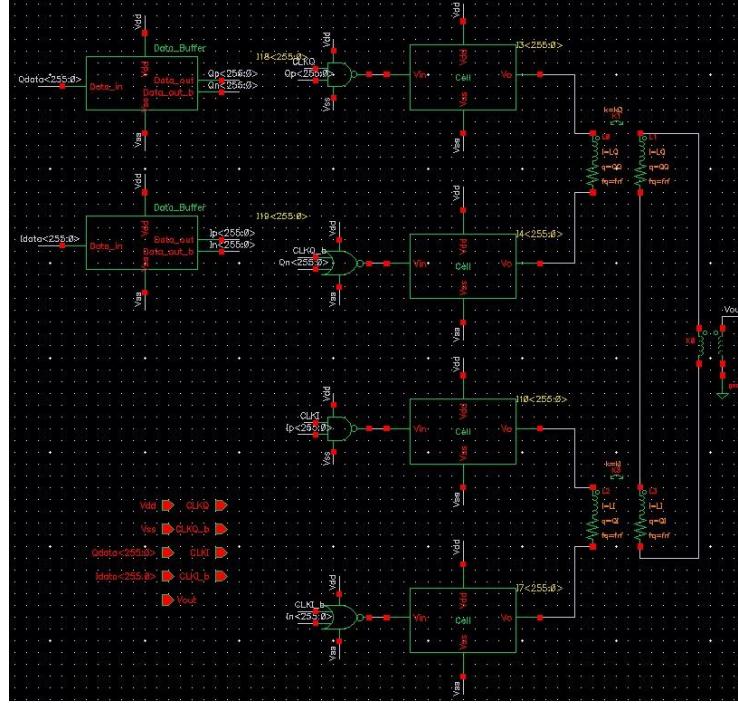
LQ	LI	One channel Vout_h1	One channel Pout_h1	Two channel Vout_h1	Two channel Pout_h1
Filter	Filter	Filter	Filter	Filter	Filter
1n	1n	2.272	17.13	3.533	20.96
1.05n	1.05n	2.351	17.42	3.485	20.84
1.1n	1.1n	2.409	17.64	3.412	20.66
1.15n	1.15n	2.428	17.7	3.353	20.51
1.2n	1.2n	2.413	17.65	3.305	20.38
1.25n	1.25n	2.378	17.53	3.252	20.24
1.3n	1.3n	2.333	17.36	3.193	20.09
1.35n	1.35n	2.282	17.17	3.13	19.91
1.4n	1.4n	2.23	16.97	3.065	19.73
1.45n	1.45n	2.178	16.76	3	19.54
1.5n	1.5n	2.128	16.56	2.936	19.36

The lower the inductance the higher the output power, when both channels are on. However, when only one channel is on, the output power gets a maximum for $L = 1.15nH$. As discussed before, ideally the output power, when two channels are on, should be $3.01dBm$ larger than when only one channel is on. However, due to nonlinear effects the output power is different. For $L = 1.5nH$ we get $2.8dBm$ difference, but for $L = 1nH$ we get $3.83dBm$ difference. If we kept decreasing the inductance, the output power would begin to decrease.

256 bit Differential IQ SCPA

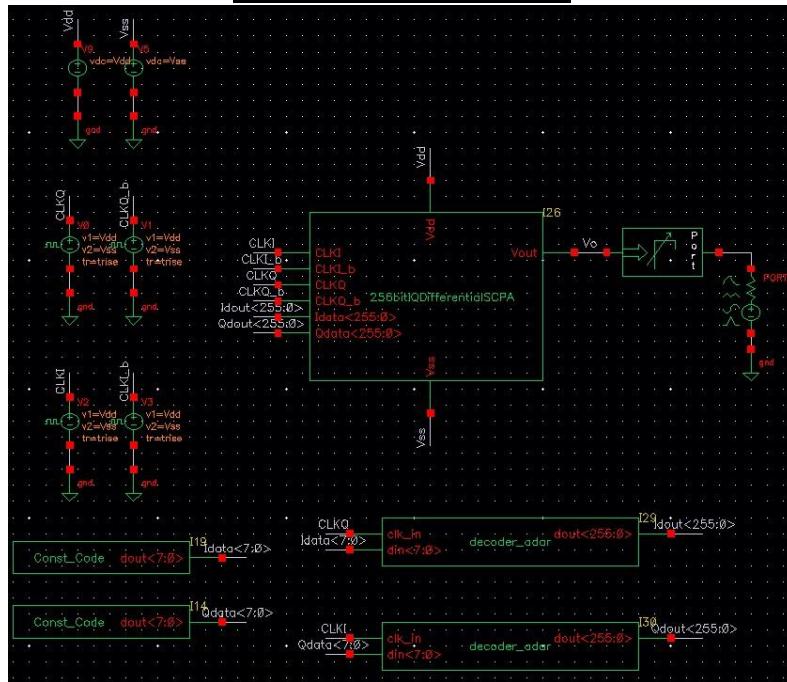
We analyze the behavior of the circuit by extracting its IQ constellation diagram and its output's amplitude, phase and power as a function of the input code.

256 bit differential IQ SCPA



To control how many cells are on in each channel, we connect the NAND and NOR mixers to the cells. Additionally, we use Data_Buffer cells as before, to invert the data. The mixers' and Data_Buffers' transistors are large enough to push the capacitance they are connected to.

256 bit SCPA – Test Bench

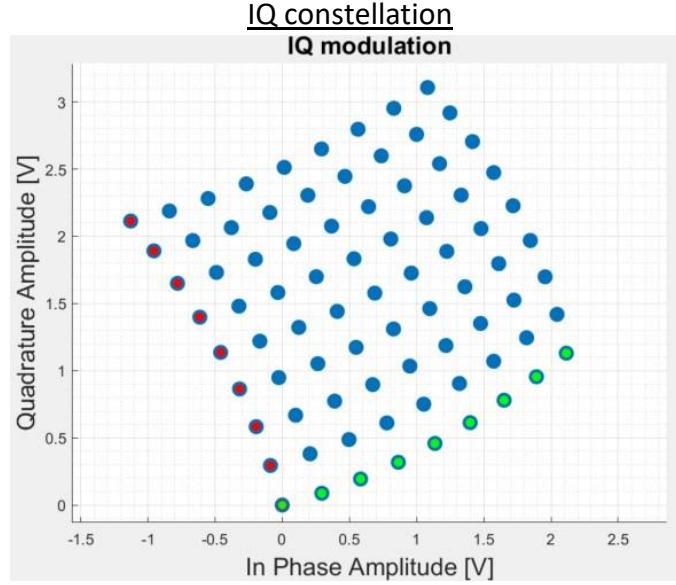


To control the input bit code, in the simulation, we use Const_Code as before. To translate the 8bit binary code to 256bit thermometric code, we use a self-made VerilogA decoder component.

Decoder VerilogA code

<pre> `include "constants.vams" `include "disciplines.vams" module decoder_adar(dout, clk_in, din); output [255:0] dout; electrical [255:0] dout; input clk_in; electrical clk_in; input [7:0] din; electrical [7:0] din; parameter real trise = 0 from [0:inf]; parameter real tfall = 0 from [0:inf]; parameter real tdel = 0 from [0:inf]; parameter real vlogic_high = 1.2; parameter real vlogic_low = 0; parameter real vtrans_clk = 0.6; real vout[255:0]; real polarity; genvar input_bit_index; integer j; integer i; genvar k; `define INPUT_BITS 8 `define OUTPUT_BITS 255 </pre>	<pre> analog begin @(cross(V(clk_in) - vtrans_clk, 1)) begin i = 0; for (input_bit_index = 0; input_bit_index < `INPUT_BITS ; input_bit_index = input_bit_index + 1) begin if(V(din[input_bit_index]) > vtrans_clk) begin polarity = vlogic_high; end else begin polarity = vlogic_low; end for(j = i; j < i + 2 ** input_bit_index; j = j + 1) begin vout[j] = polarity; end i = i + 2 ** input_bit_index; end for(k = 0; k <= `OUTPUT_BITS; k = k + 1) begin V(dout[k]) <+ transition(vout[k], tdel, trise, tfall); end `undef INPUT_BITS `undef OUTPUT_BITS end endmodule </pre>
--	---

As done in the IQ modulation section, we extract the IQ constellation diagram. For clarity, we didn't turn on cells one by one, but in chunks 0:32:224, 225.

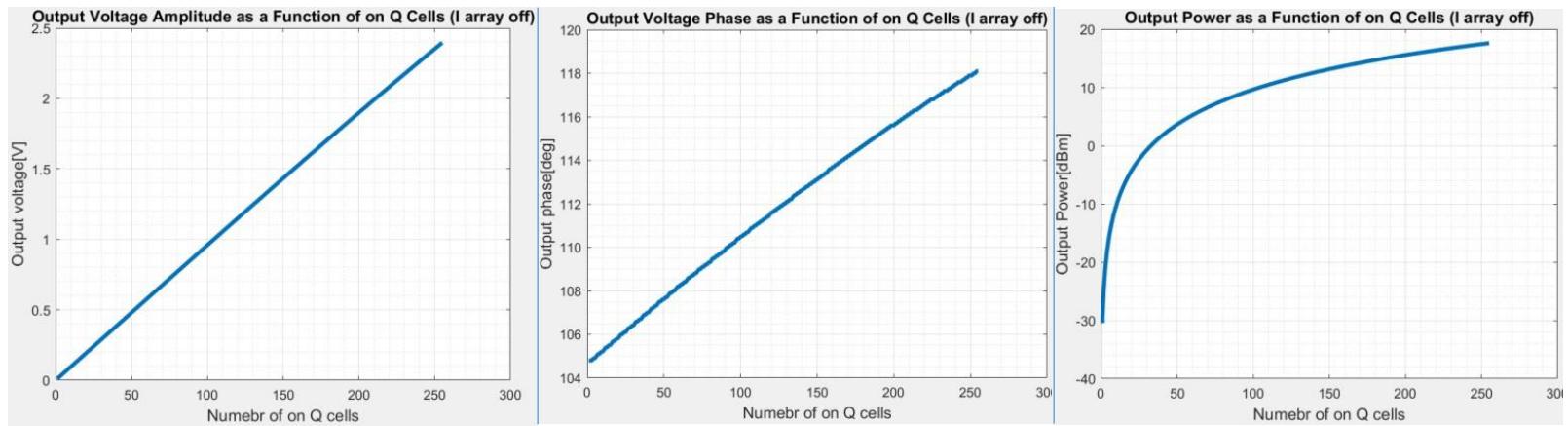


The red and green points refer to cases where the I and Q arrays (channels) are off, respectively. These points are approximately orthogonal. Our setup lets us transmit data only in quarter of the constellation diagram. We can transmit data in other quarters by inverting the clocks. We do that by using two additional bits. Each bit controls whether a clock should be inverted or not. The non-linearity of the circuit is seen in the curvature of the constellation.

Next, we plot the output's amplitude phase and power of the fundamental frequency, as a function of how many cells are on. We denote the number of cells on in the Q and I channels as n_Q and n_I , respectively. We denote ϕ_o as the offset phase of the IQ constellation. Thus, the output voltage phase should be $\angle V_{out} = \tan^{-1}\left(\frac{n_Q}{n_I}\right) + \phi_o$. Additionally, we denote the output voltage amplitude, each cell contributes to its channel's component, as v_o . Thus, the output voltage amplitude should be $|V_{out}| = v_o \sqrt{n_Q^2 + n_I^2}$. The output power should be $P_{out} = 10 \log(\alpha |V_{out}|^2)$, where α is some constant. Thus, $P_{out} = 10 \log(\alpha v_o^2) + 10 \log(n_Q^2 + n_I^2)$.

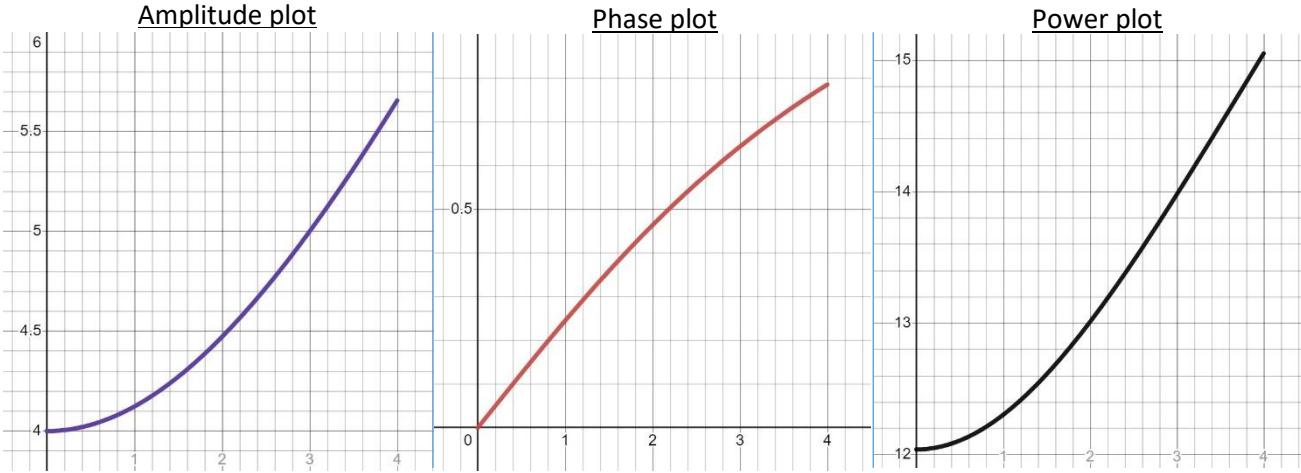
We begin with a simulation where the I channel is off and we turn on the Q channel's cells. In this case, the amplitude should look like $|V_{out}| = v_o n_Q$, the phase like $\angle V_{out} = 90^\circ + \phi_o$ and the power $P_{out} = 10 \log(\alpha v_o^2) + 20 \log(n_Q)$.

Amplitude, Phase and Power when the I channel is off



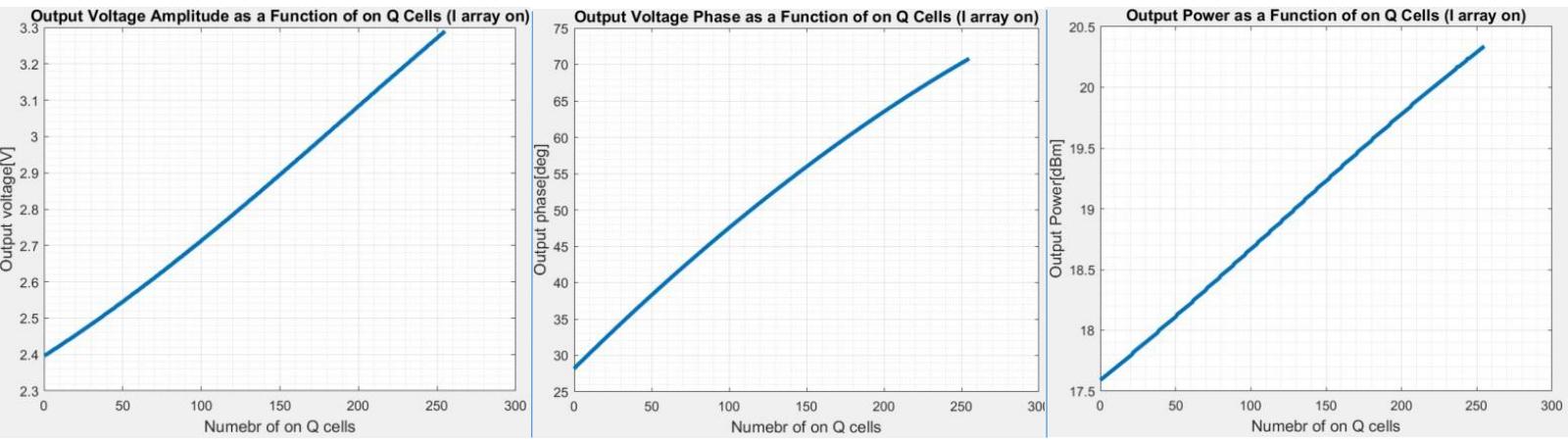
The amplitude graph looks about right. It is linear and begins at 0V. The phase graph isn't constant, as in the ideal case, due to the circuit's non-linear behavior. The power graph looks logarithmic, as it should.

Now, we simulate when the I channel is fully on and we turn on the Q channel's cells. In this case, the amplitude should look like $|V_{out}| = v_o \sqrt{n_Q^2 + 256^2}$, the phase like $\angle V_{out} = \tan^{-1} \left(\frac{n_Q}{256} \right) + \phi_o$ and the power $P_{out} = 10 \log(\alpha v_o^2) + 10 \log(n_Q^2 + 256^2)$. To understand better how these graphs should look, we plot them for $n_I = 4$, $v_o = 1$, $\phi_o = 0^\circ$ and $\alpha = 1$.



The amplitude and power graphs are saturated at the beginning. They grow, but at some point become relatively straight. The phase graph is a little concave (curved inwards).

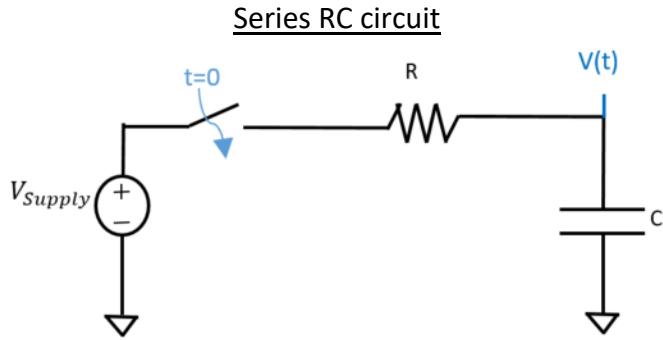
Amplitude, Phase and Power when the I channel is fully on



Both the amplitude and power graphs are relatively straight, but they are not saturated at the beginning. This behavior is explained by the IQ constellation. Because it is curved and not straight, like in the ideal case, the amplitude increases faster and there is no saturation zone. The phase graph looks a little concave, as it should.

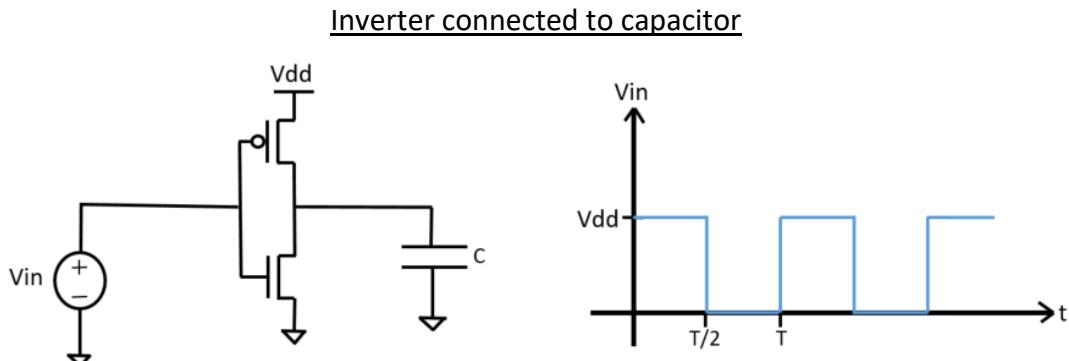
Sizing Logic Gates

An electronic circuit is usually filled with capacitors. Some capacitors are purposely placed, like in the case of SC DAC, and some are intrinsic, from the physical structure of transistors. As these capacitors charge and discharge, they effect the circuit's behavior. The simplest case is a capacitor C charging and discharging through a constant resistor R .



A series RC circuit is connected at time $t = 0$ to a constant voltage supply V_{supply} . We denote the initial capacitor voltage as $V(0)$, the final capacitor voltage as $V(\infty) = V_{supply}$ and its voltage at time t as $V(t)$. From first order differential equation, we can derive that the time it takes for the capacitor to charge or discharge to some voltage $V(t)$ is $t = RCLn\left(\frac{V(\infty)-V(0)}{V(\infty)-V(t)}\right)$.

We can use this circuit to model the behavior of logic gates. The simplest case is an inverter connected to a load capacitor C . The capacitor is either purposely placed, or the total intrinsic input capacitance of the next gates.



Let's say the inverter's input is a rectangular voltage waveform as shown in the graph. Both the supply voltage and the input voltage have an amplitude of Vdd . When $Vin = Vdd$ only the NMOS is open and the capacitor is discharging. When $Vin = 0$ only the PMOS is open and the capacitor is charging. Using the effective resistance of the transistors, when they are open, we can model this circuit as the RC circuit above, where the transistors are both the switch and resistor. When the PMOS is open $V(\infty) =$

$V_{supply} = V_{dd}$, $V(0) = 0$. When the NMOS is open $V(\infty) = V_{supply} = 0$, $V(0) = V_{dd}$. This is under the assumption that the capacitor has enough time to almost charge or discharge completely between switches. When logic gate charges and discharges a capacitor, we say it pushes it.

The intrinsic input capacitance of a gate C is proportional to the gate's size. The effective resistance of transistors R is inversely proportional to the gate's size. The effective resistance is usually different between NMOS and PMOS, and is dependent on the input. However, for simplicity we assume it is a constant value.

As we saw, the charge and discharge time of a capacitor is proportional to RC . The larger a gate, the smaller its R and the faster it pushes its load capacitance. If the load capacitor is the total input intrinsic capacitance of the next gates, the smaller the gates the smaller the load C and faster the charge and discharge time.

An important measure is the rise and fall time at the input of logic gates. Any logic gate has a capacitance at its input. The rise (fall) time is the time it takes the capacitor to charge (discharge) from 10% to 90% (90% to 10%). If we denote $\Delta V = V(\infty) - V(0)$, the rise time is measured from $V(0) + 0.1\Delta V$ to $V(0) + 0.9\Delta V$ and the fall time from $V(\infty) + 0.9\Delta V$ to $V(\infty) + 0.1\Delta V$.

Rise and fall time are important for two main reasons. Firstly, we want to reduce the feedthrough current, which is a big consumer of power. The longer the rise/fall time, the longer both PUN and PDN transistors are open and the more power consumed. Secondly, the frequency of the signal propagating through the circuit might increase (frequency modulation). In this case, the signal's width can get relatively small, so we want the signal's rise/fall times to be relatively small as well.

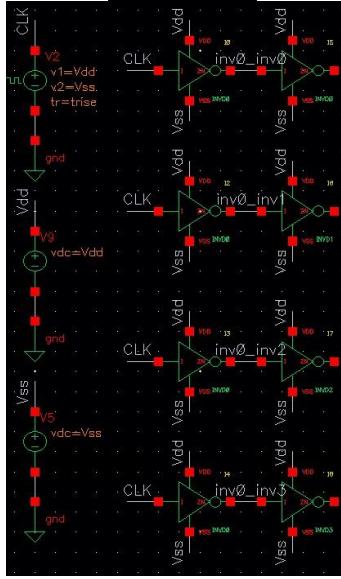
It is important to size logic gates correctly, so the rise and fall time at their inputs is short. Large gates' transistors have smaller effective resistance, thus pushing their load capacitance faster and causing fast rise and fall time at the next gate's input. On the other hand, large gates have large intrinsic input capacitance, so it is harder for other gates to push them. Small gates have small intrinsic input capacitance, so it would be easier for other gates to push it. However, it is harder for small gates to push a load. Moreover, slow transition at the input of a gate causes its transistors to switch slower, so its load capacitance charges and discharges slower, increasing the rise and fall time.

Additional consideration is the dynamic power consumed by large gates. The dynamic power consumption of logic gates results from the switching of the gate and the charge and discharge of its load capacitor. The average dynamic power consumption is $P_{dynamic} = CV^2f$. Thus, large gates with large intrinsic input capacitance cause large dynamic power consumption.

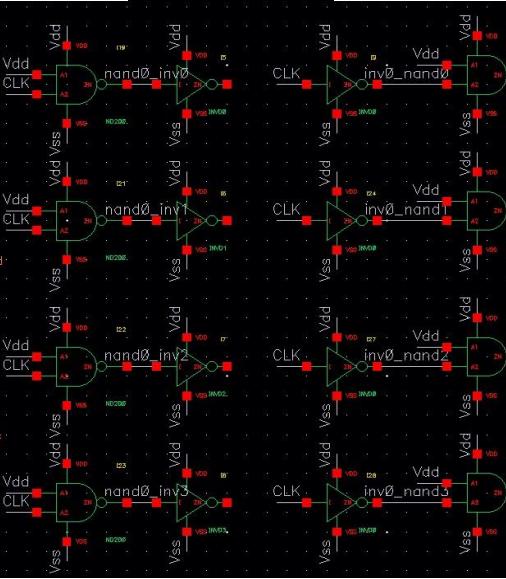
We denote fanout as of a gate as $f = \frac{C_{ext}}{C_{gin}}$, where C_{ext} is the load capacitance of the gate and C_{gin} is the intrinsic input capacitance of the gate. Usually, gates with equal fanout and input transition, have the same rise and fall time for their loads. For example, INVD1 with INVD2 as load has the same fanout as INVD2 with INVD4 as load. Thus, we can expect their load's rise/fall time to be the same.

We simulate the rise and fall time of different gate size and order. The results should be the same for gates with the same order but with larger gates. The gates we use in our design are inverters, NANDs and NORs, so we simulate them. For realistic response we set the rise and fall time of the clock to 30ps (between 10% and 90%). The minimal sized gate is half a finger. It means that the transistors have a single finger but half the channel width.

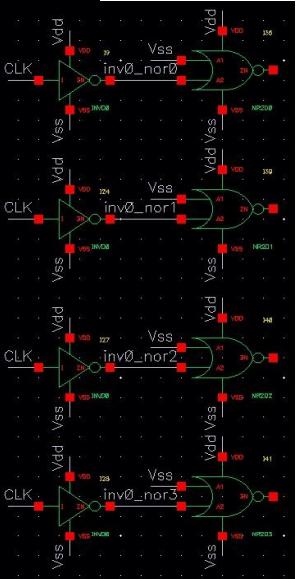
Inverter test



NAND test bench



NOR test bench



Simulation results

Inverter results

inv0_inv0 rise	17.83p
inv0_inv1 rise	24.4p
inv0_inv2 rise	38.28p
inv0_inv3 rise	52.61p
inv0_inv0 fall	13.15p
inv0_inv1 fall	18.26p
inv0_inv2 fall	28.81p

NAND results

nand0_inv0 rise	22.63p
nand0_inv1 rise	29.6p
nand0_inv2 rise	43.9p
nand0_inv3 rise	58.4p
inv0_nand0 rise	15.57p
inv0_nand1 rise	22.01p
inv0_nand2 rise	39.65p
inv0_nand3 rise	54.82p
nand0_inv0 fall	23.63p
nand0_inv1 fall	33.01p
nand0_inv2 fall	51.98p
nand0_inv3 fall	71.07p
inv0_nand0 fall	12.31p
inv0_nand1 fall	15.93p
inv0_nand2 fall	25.99p
inv0_nand3 fall	37.54p

NOR results

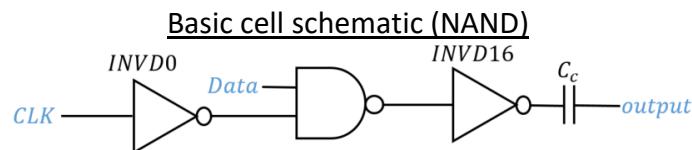
nor0_inv0 rise	33.65p
nor1_inv0 rise	26.81p
inv0_nor0 rise	15.57p
inv0_nor1 rise	22.18p
inv0_nor2 rise	37.43p
inv0_nor3 rise	50.81p
nor0_inv0 fall	16.68p
nor1_inv0 fall	13.75p
inv0_nor0 fall	12.24p
inv0_nor1 fall	15.77p
inv0_nor2 fall	24.77p
inv0_nor3 fall	34.89p

X_Y means that gate X is the input gate and gate Y is the load gate. In our design, we constrain the rise and fall time to be smaller than $40ps$, from which we derive the maximum fanout of a gate. We conclude the following:

- Inverter can push another inverter 4 times its size
- Inverter can push NAND 4 times its size
- Inverter can push NOR 4 times its size
- NAND can push an inverter 2 times its size
- NOR can push an inverter 1 times its size

Sizing the Unit Cell

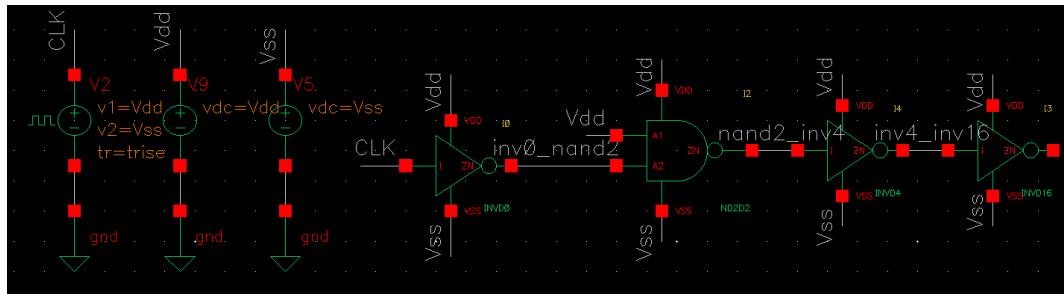
Our design's SC DAC uses INVD16 inverters to push the capacitors. INVD16 is 32 times larger than the minimal sized inverter and has 16 fingers. We have 256 cells in parallel, which we need to distribute the clock to. The clock sees 256 cells in parallel, which is 256 times the input capacitance of a single cell. If this capacitance is too large, the rise and fall time at the input of the cells will be very bad. Thus, we want the input capacitance of a single cell to be minimal, so we connect the minimal sized inverter INVDO to the input. INVDO has half a finger, meaning its transistors' channel width is half. Between INVDO and INVD16 we have a NAND/NOR mixer.



From the logic gate sizing section, we know that no NAND size will be sufficient to achieve less than $40ps$ rise and fall time. Thus, we need to add inverters. We add them after the NAND gate, so when the NAND isn't switching they won't as well, saving dynamic power.

Inverter can push NAND 4 times its size, so we pick ND2D2. We need to add inverters between the NAND and INVD16. NAND can push inverter 2 times its size, so we pick INVDO. Inverter can push another inverter 4 times its size, so it seems INVDO can push INVD16.

NAND Unit Cell 1

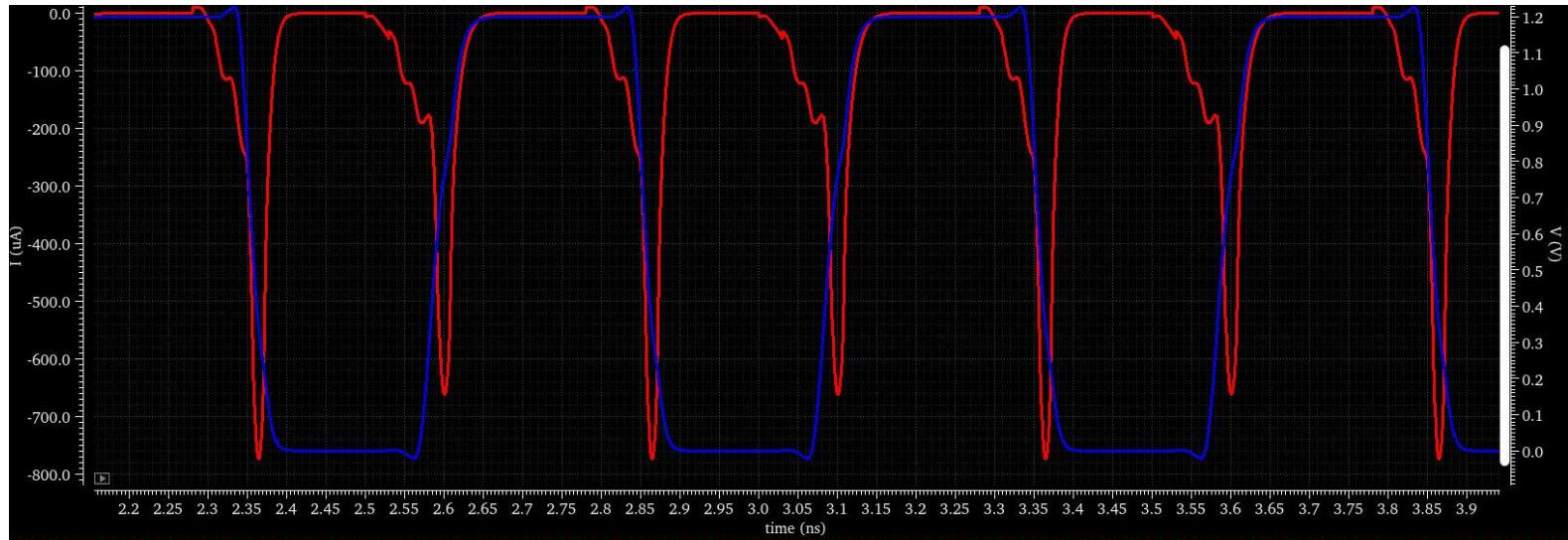


Simulation results

inv0_nand2 rise	34.23p
nand2_inv4 rise	30.68p
inv4_inv16 rise	45.62p
inv0_nand2 fall	23.94p
nand2_inv4 fall	32.37p
inv4_inv16 fall	33.41p
PDC	113uW

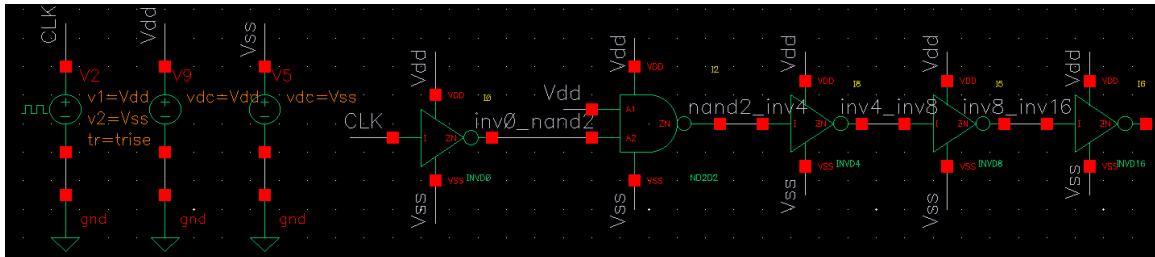
The rise time between INV4 and INV16 is too large, so we add INV8 between. It doesn't work connecting INV4 to INVD16, because the transition time at INV4's input is too slow. The average power consumption of this cell is $113\mu W$.

Unit Cell 1 – current graph



The red graph is the current flowing to the supply voltage. The blue graph is the voltage between INVD4 and INVD16. The current peaks as the voltage graph transitions. When the input of a logic gate transitions, both PUN and PDN are open, causing large feedthrough current from the upper supply voltage to the lower.

NAND Unit Cell 2



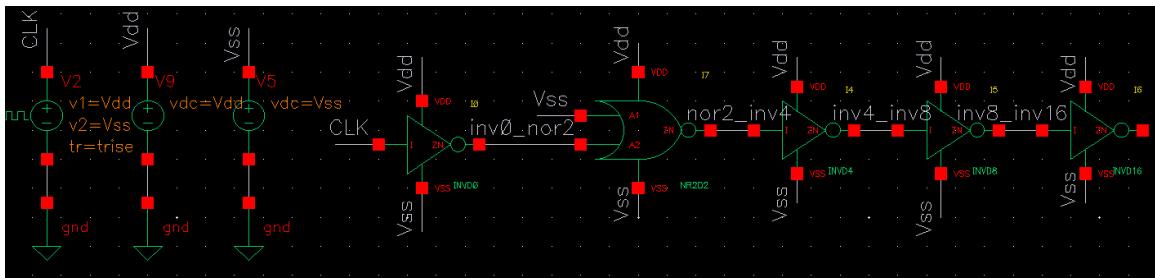
Simulation results

inv0_nand2 rise	34.23p
nand2_inv4 rise	32.57p
inv4_inv8 rise	28.94p
inv8_inv16 rise	26.32p
inv0_nand2 fall	23.84p
nand2_inv4 fall	34.18p
inv4_inv8 fall	19.62p
inv8_inv16 fall	21.46p
PDC	152.4u

All of the rise and fall times are good now. However, the average power consumption raised by a fair amount. The feedthrough current through the gates decreased, but by adding additional gate, the dynamic power consumption increased.

For the NOR unit cell, we simply replace the ND2D2 with a NOR gate of the same size NR2D2. However, we know NOR can't push an inverter twice its size, so we expect the rise time to be above 40ps.

NOR Unit Cell 1



Simulation results

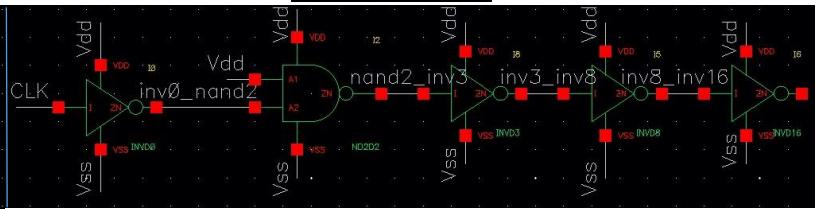
inv0_nor2 rise	34.95p
nor2_inv4 rise	52.3p
inv4_inv8 rise	25.45p
inv8_inv16 rise	27.21p
inv0_nor2 fall	24.1p
nor2_inv4 fall	24.76p
inv4_inv8 fall	24.98p
inv8_inv16 fall	20.86p
PDC	152.6u

As expected the rise time between the NOR and inverter is too high. We don't want to add more inverter, because then the power consumption will greatly increase. Instead, we change INVD4 to INVD3.

NAND Unit Cell 3



NOR Unit Cell 2



Simulation results

NAND results

inv0_nand2 rise	34.88p
nand2_inv3 rise	27.49p
inv3_inv8 rise	33.37p
inv8_inv16 rise	27.06p
inv0_nand2 fall	24.1p
nand2_inv3 fall	28.41p
inv3_inv8 fall	21.87p
inv8_inv16 fall	22.68p
PDC	147.5u

NOR results

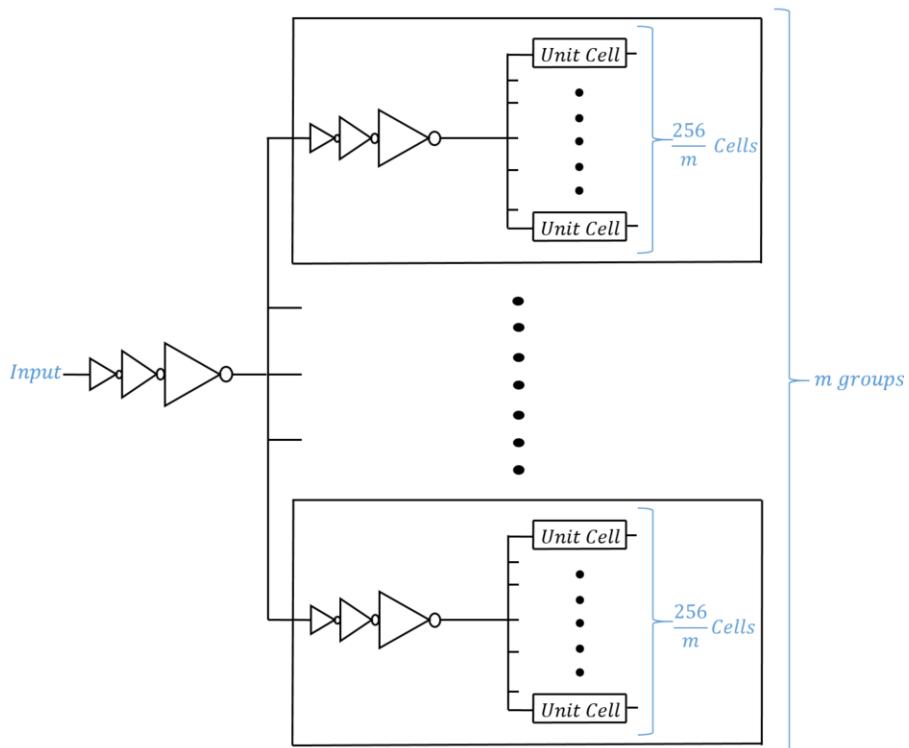
inv0_nor2 rise	35.57p
nor2_inv3 rise	43.41p
inv3_inv8 rise	30.71p
inv8_inv16 rise	27.82p
inv0_nor2 fall	24.17p
nor2_inv3 fall	21.95p
inv3_inv8 fall	26.49p
inv8_inv16 fall	22.29p
PDC	147.7u

The NAND rise and fall time are still good. The rise time between the NOR and INVD3 is better, but still a little higher than 40ps. The power consumption in both cases is a little better, because INVD3 is smaller than INVD4.

Clock Tree

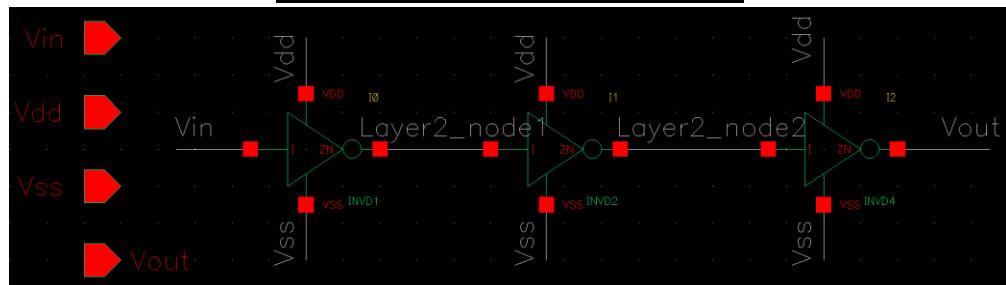
Even though we made the input capacitance of the cells minimal $C_{minimal}$, the clock sees 256 cells in parallel (layer 1) and a large capacitance $256C_{minimal}$. Thus, we divide the cells into m groups and each group we push separately, using inverter chains (layer 2). Each chain needs to push $\frac{256}{m}C_{minimal}$. If the chains' first inverter is minimal, the clock will need to push $mC_{minimal}$. We want the clock to see the minimal possible capacitance, so its transition time will be ideal. Thus, we push the m chains with another chain (layer 3) and set its first inverter to be minimal.

Three Layers Clock Tree Schematic

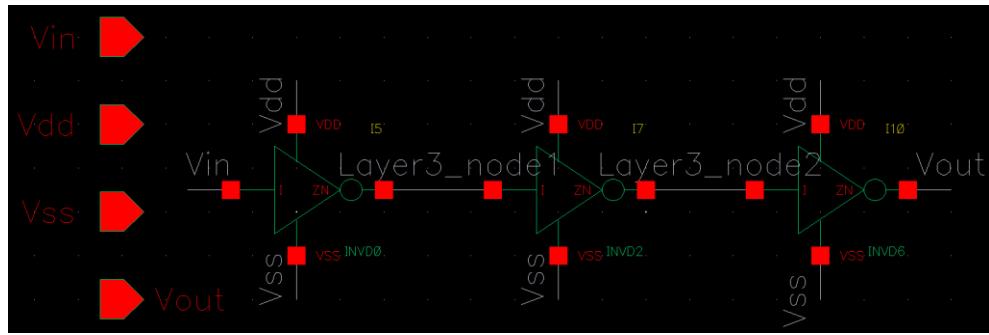


For layers 2 and 3 to push the same capacitance we set $m = 16$. In this case, each chain pushes capacitance equals to $16C_{minimal}$, which is the same as the intrinsic input capacitance of INVD8. Instead of taking the first inverter of layer 2 to be minimal sized, we take INVD1 and make layer 3 inverter chain push harder.

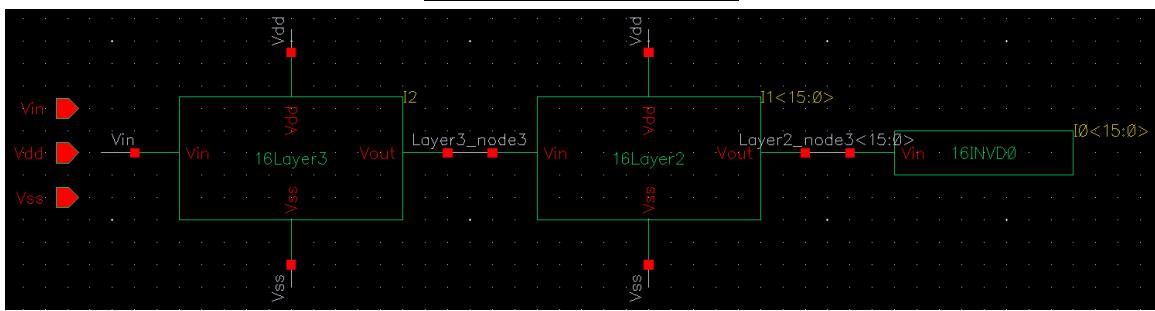
Layer 2 Inverter chain – schematic 1



Layer 3 Inverter chain – schematic 1



Clock Tree - schematic 1

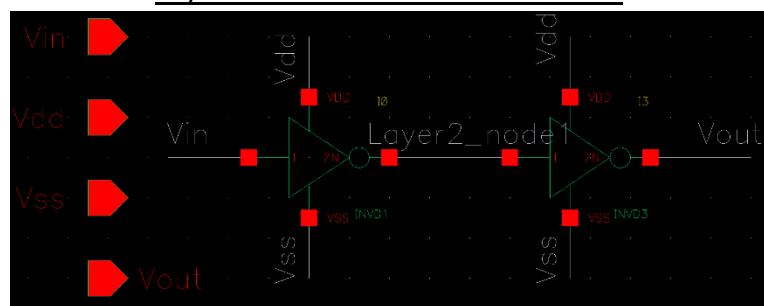


Simulation results 1

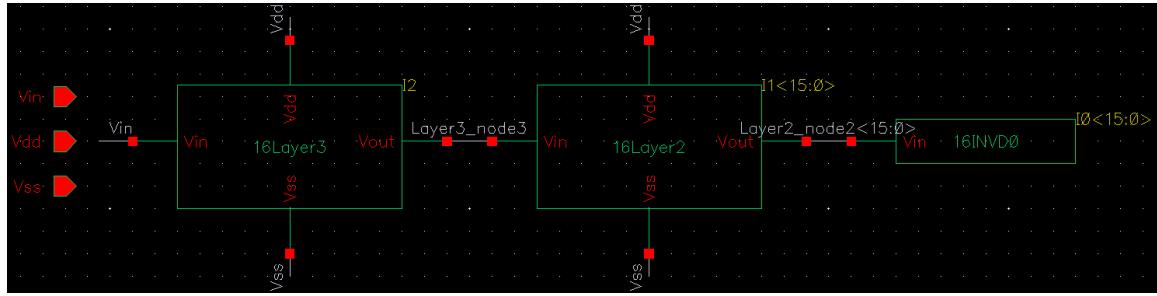
Layer3_node1 rise	37.51p
Layer3_node2 rise	34.87p
Layer3_node3 rise	32.76p
Layer2_node1 rise	25.87p
Layer2_node2 rise	24.5p
Layer2_node3 rise	26.92p
Layer3_node1 fall	24.4p
Layer3_node2 fall	26.26p
Layer3_node3 fall	24.19p
Layer2_node1 fall	19.79p
Layer2_node2 fall	18.3p
Layer2_node3 fall	21.35p
PDC	961.2u

Because the first inverter of the second layer isn't minimal sized, it has only three inverters. All of the rise and fall times are less than 40ps and the power consumption is 961.2uW. The rise and fall time of the second layer is very good. We can trade it to reduce the power consumption. We use the same third layer, but switch INV2 and INV4, in the second layer, with INV3.

Layer 2 Inverter chain – schematic 2



Clock Tree - schematic 2



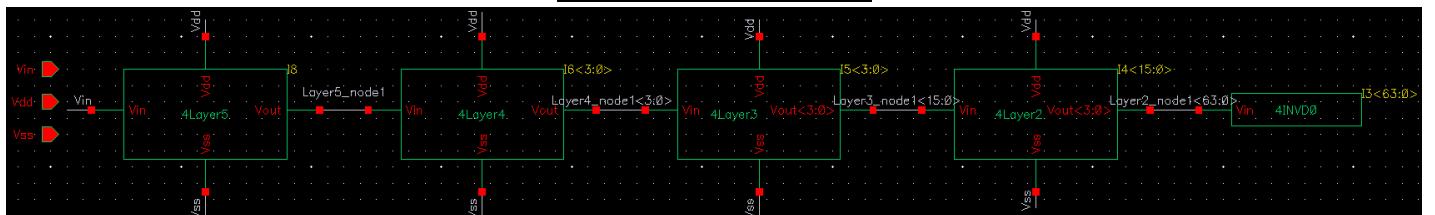
Simulation results 2

Layer3_node1 rise	37.51p
Layer3_node2 rise	34.9p
Layer3_node3 rise	30.97p
Layer2_node1 rise	33.44p
Layer2_node2 rise	33.77p
Layer3_node1 fall	24.38p
Layer3_node2 fall	26.27p
Layer3_node3 fall	23.58p
Layer2_node1 fall	24.36p
Layer2_node2 fall	27.76p
PDC	721.3u

The power consumption is now $721.3\mu W$, more than $200\mu W$ better than before, at the cost of slower rise and fall time in the second layer. It must be noted that it is possible to reduce the second layer to only INVD3, by adding INVD16 at the end of the third layer. However, the results won't change. It makes sense because 16 INVD1 is parallel have the same capacitance as INVD16.

Another option is dividing the cells into 64 groups of 4 and adding layers. The initial thought is having a single inverter INVD0 in each layer and having each INVD0 push 4 INVD0. In this case, there will be 5 layers. Layers 2, 3, 4 and 5 will have 64, 16, 4 and 1 INVD0 in parallel, respectively.

Clock Tree - schematic 3

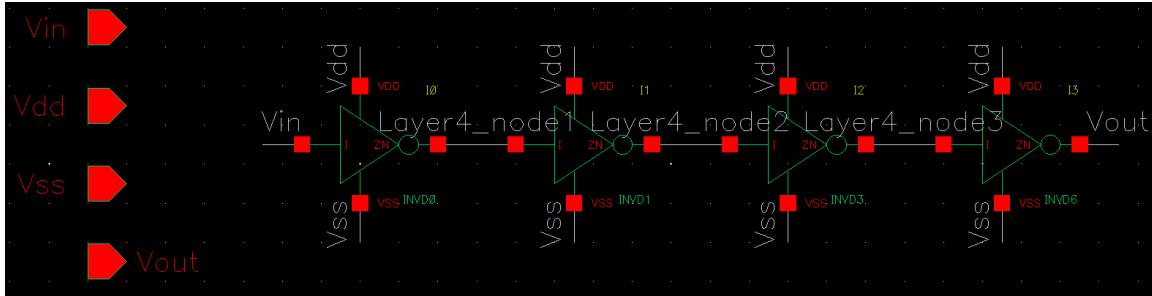


Simulation results 3

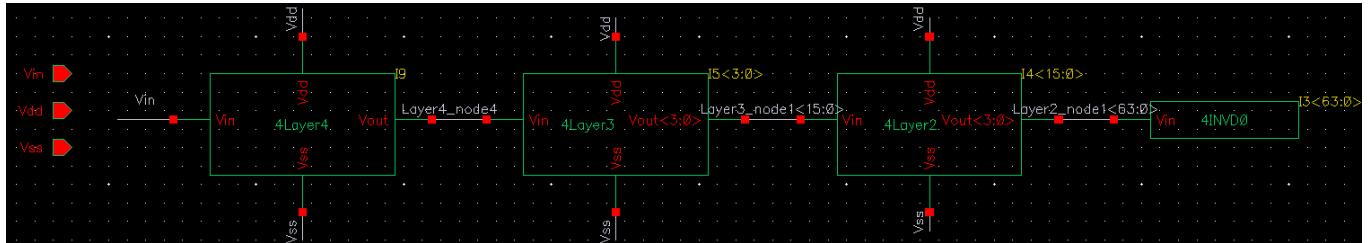
Layer5_node1 rise	37.31p
Layer4_node1 rise	40.63p
Layer3_node1 rise	41.17p
Layer2_node1 rise	43.46p
Layer5_node1 fall	24.66p
Layer4_node1 fall	30.82p
Layer3_node1 fall	31.53p
Layer2_node1 fall	36.15p
PDC	585.2u

The power consumption is better, but some of the rise times are larger than 40ps.
Again, we can decrease the rise time at the cost of power. Instead of using 5 layers, we use 4 layers, with longer layer 4. Additionally, we switch layer 3 inverters with INV1, to decrease the rise time at the input of the second layer.

Layer 4 Inverter chain – schematic



Clock Tree - schematic 4



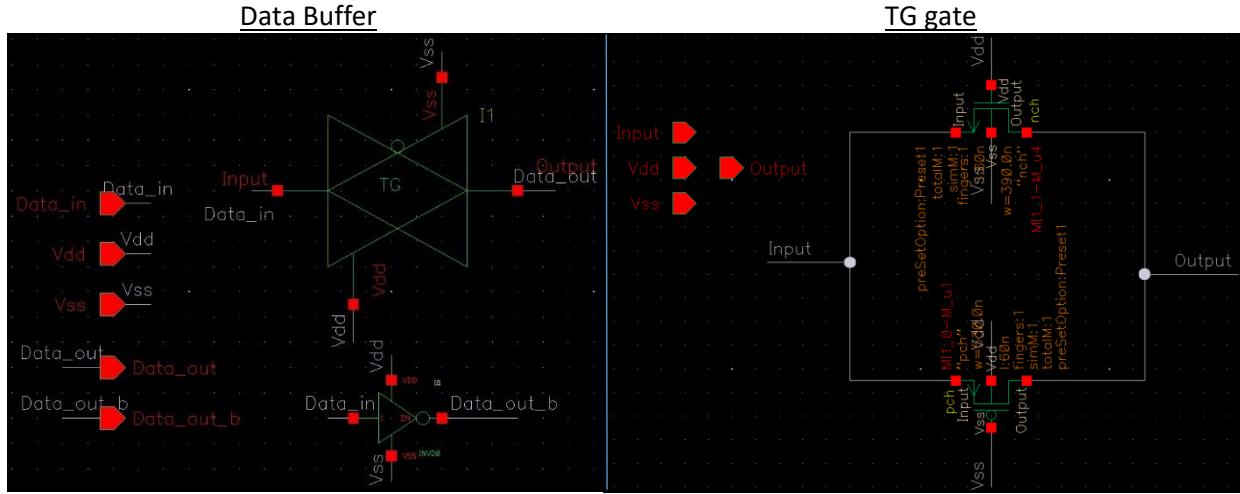
Simulation results 4

Layer4_node1 rise	21.28p
Layer4_node2 rise	33.71p
Layer4_node3 rise	25.03p
Layer4_node4 rise	30.97p
Layer3_node1 rise	24.41p
Layer2_node1 rise	40.69p
Layer4_node1 fall	15.68p
Layer4_node2 fall	22.57p
Layer4_node3 fall	20.1p
Layer4_node4 fall	21.54p
Layer3_node1 fall	19.62p
Layer2_node1 fall	32.16p
PDC	664.3u

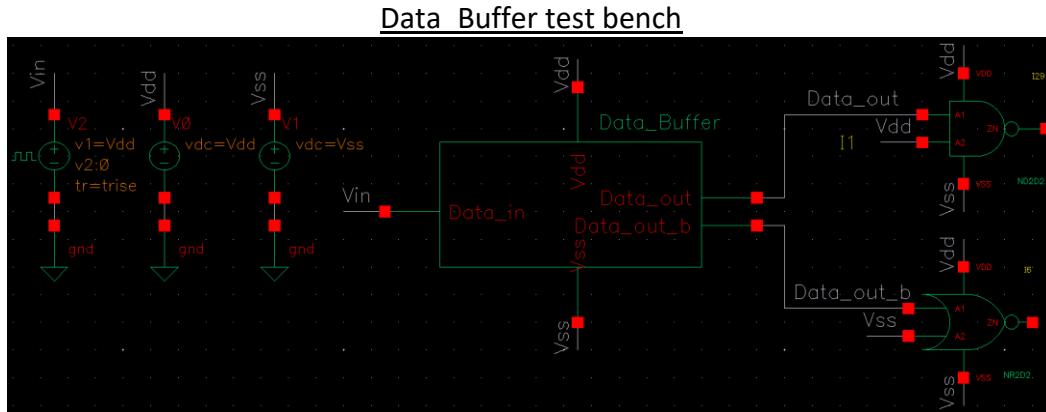
The rise time is better, but the rise time between the first and second layer is still a little above 40ps. The power consumption increased, but it is still better than the first two suggested clock trees.

Data Buffer and Mux

Previously, we used a self-made Data_Buffer cell, to get the inverted value of the data. On one hand, the data goes into an inverter, to be inverted. On the other hand, it goes through a TG switch, so the non-inverted signal will go through about the same delay as the inverted signal. The inverter is minimal sized and the TG switch is self-made out of single finger transistors.



We simulate the Data Buffer. The non-inverted output goes to a ND2D2 gate and the inverted output goes to a NR2D2 gate. Because the clock doesn't go through this gate, we don't restrict the rise and fall time to be less than $40ps$. However, we still look at the rise and fall time, as an indication to the circuit's behavior.

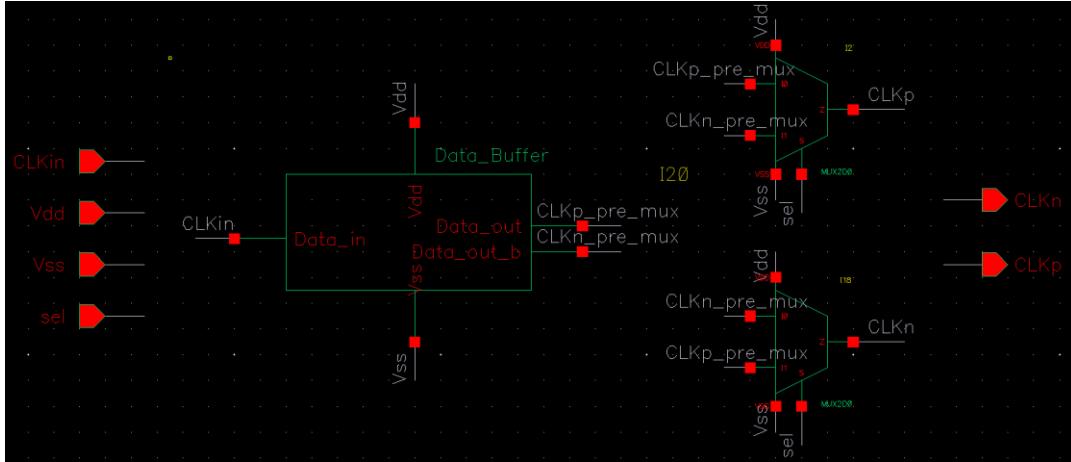


Data Buffer simulation results

Data_out_b rise	38.37p
Data_out rise	20.65p
Data_out_b fall	23.01p
Data_out fall	14.6p

In order to transmit on all four quarters of the IQ constellation space, we receive a control bit for each channel. It controls whether the clock is inverted or not. It does so by controlling a MUX selection.

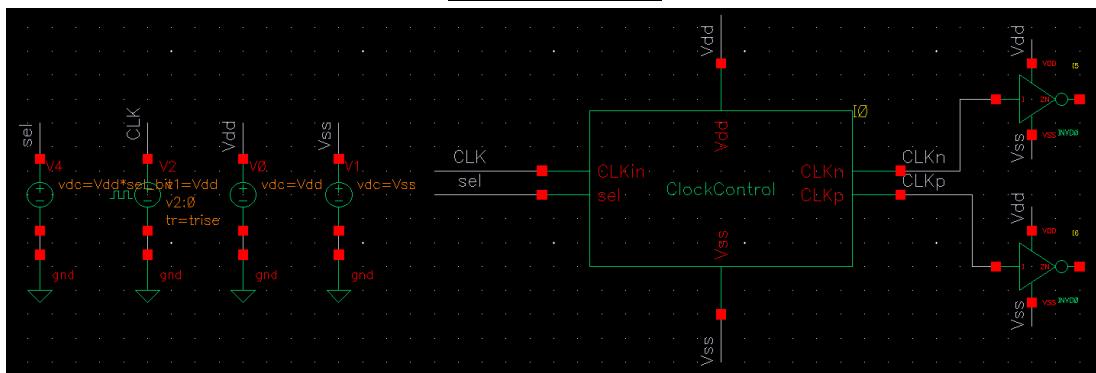
Clock Control schematic



Anyway, we need to generate the invert of the clock for the differential SCPA. We use a Data_Buffer to do so. Using the control bit, we control whether to switch between the inverted and non-inverted clocks, which is the same as inverting the clock.

We simulate the Clock Control. Each of the Clock Control's output clocks is sent to the unit cells through a clock tree. The input to the clock tree is INVD0. We need to make sure the rise and fall time are less than 40ps.

Clock Control TB



Simulation results

CLKp_pre_mux rise	21.52p
CLKn_pre_mux rise	18.92p
CLKp rise	20.81p
CLKn rise	20.72p
CLKp_pre_mux fall	18.36p
CLKn_pre_mux fall	13.8p
CLKp fall	18.95p
CLKn fall	19.02p

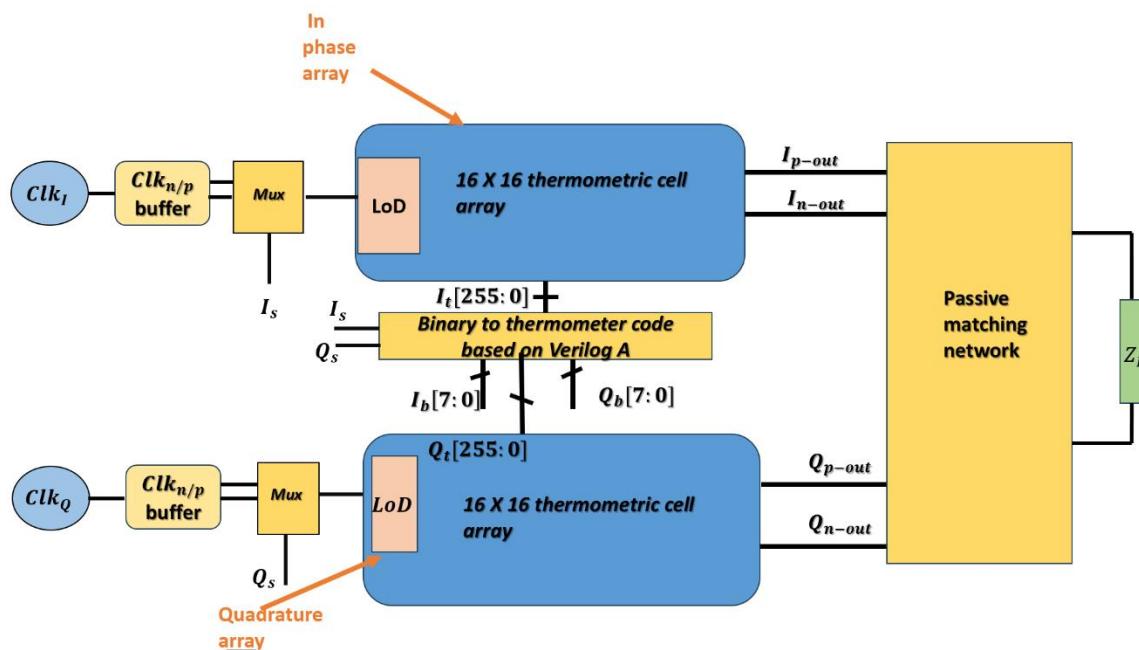
Q-SCPA

Our design is the output stage of a RF transmitter, the power amplifier (PA). It is a switch capacitor PA (SCPA), made of SC DAC combined with a matching network. The SC DAC is an array made of 256 differential unit cells, which give better common noise immunity, higher output swing and virtual ground. The matching network is made of non-ideal transformers, that are designed to transfer enough power to the load antenna.

The circuit receives an 8-bit binary input code that is translated, by a decoder, to a 256-bit thermometric code. The thermometric code determines, using logic gates, how many cells are on in an array. An on cell convey a 2GHz clock to its capacitor. The clock is distributed to all the cells through a local oscillator distribution (LoD), which is a clock tree. The clock tree's and unit cells' logic gates are sized so the rise and fall time, between the gates, won't be too high.

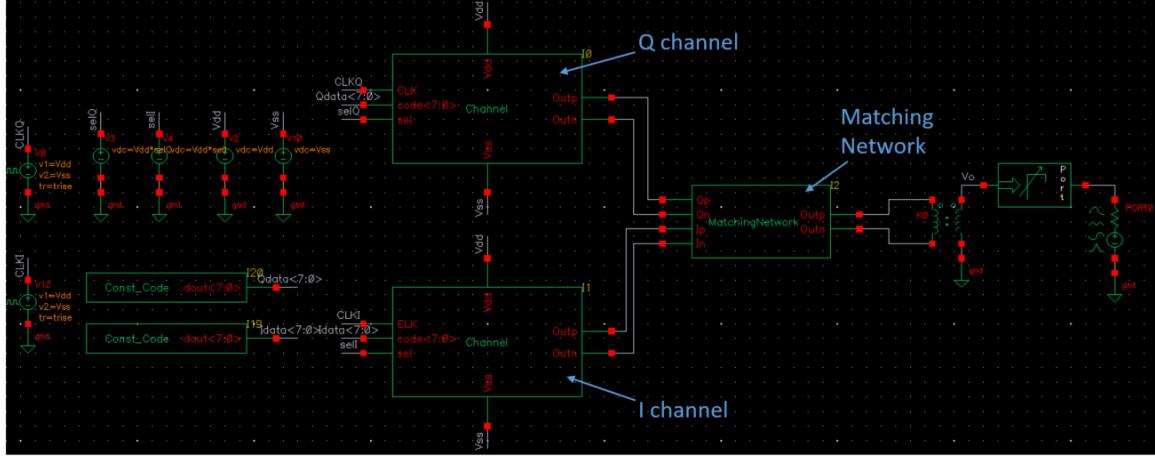
The design transmits in QAM IQ modulation, making it a Quadrature SCPA (Q-SCPA). It has two parallel SCPA channels with orthogonal clocks, which are controlled separately. The channels outputs are combined by series non-ideal transformers, that are also part of the matching network. In order to transmit on all four quarters of the IQ constellation space, each channel receives an additional bit that can invert its clock.

Q-SCPA Block Diagram

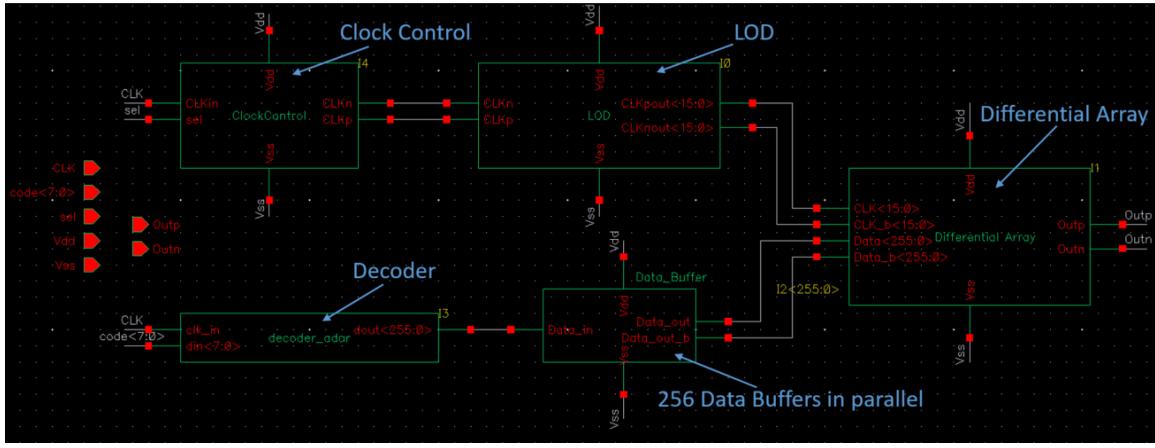


We simulate a full QSCPA. We picked the last shown Unit cells and the second shown Clock Tree. We mainly interested in the power efficiency of the circuit. That is how much power our circuit generates compare to how much it consumes. The first harmonic output power is the power the circuit generates. The parameters we use are $L = 1.2nH$, $C_c = 50fF$, $k = 0.8$ and $Q = 14$.

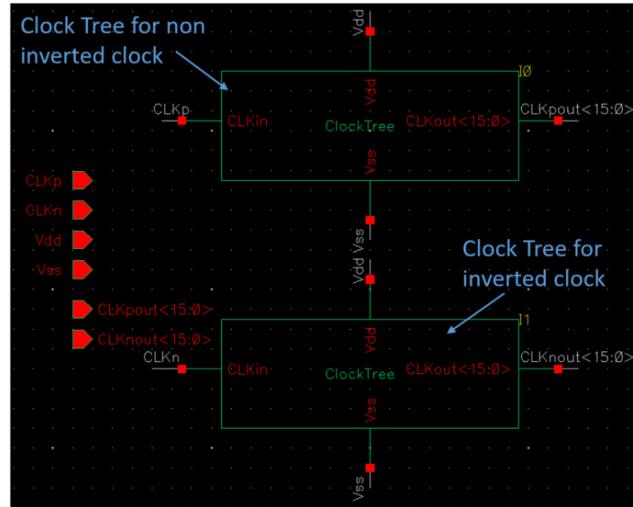
QSCPA schematic



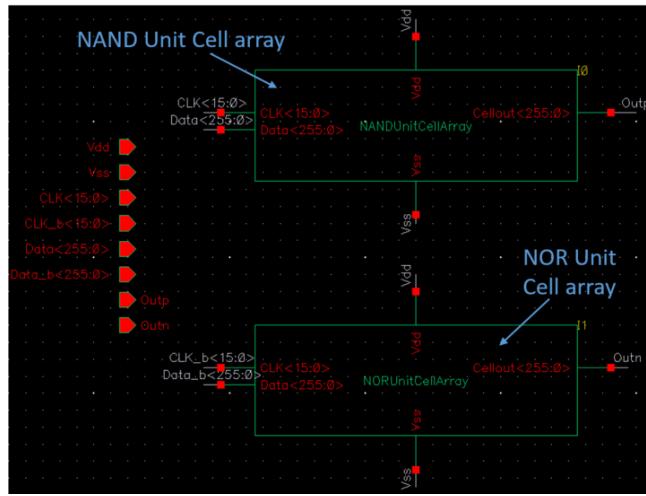
Channel schematic



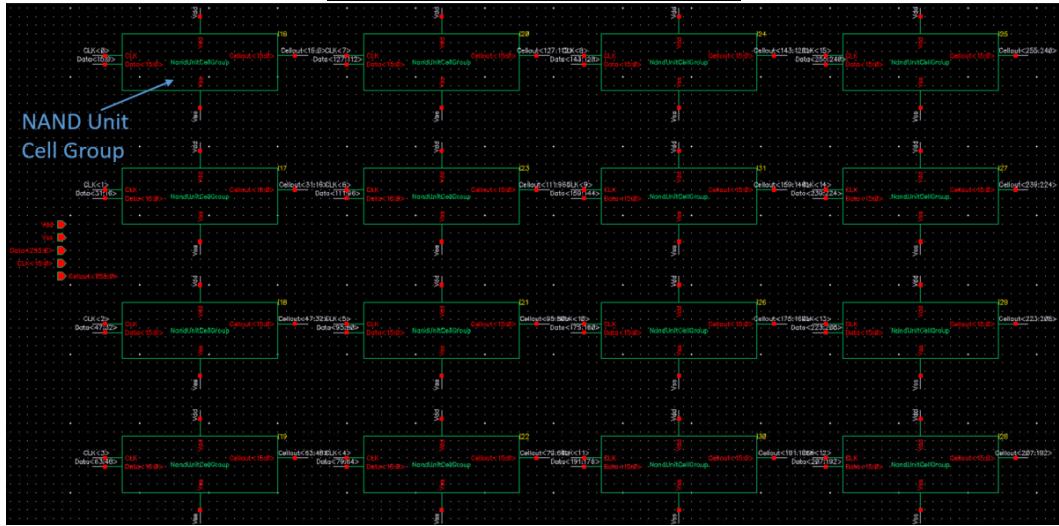
LOD schematic



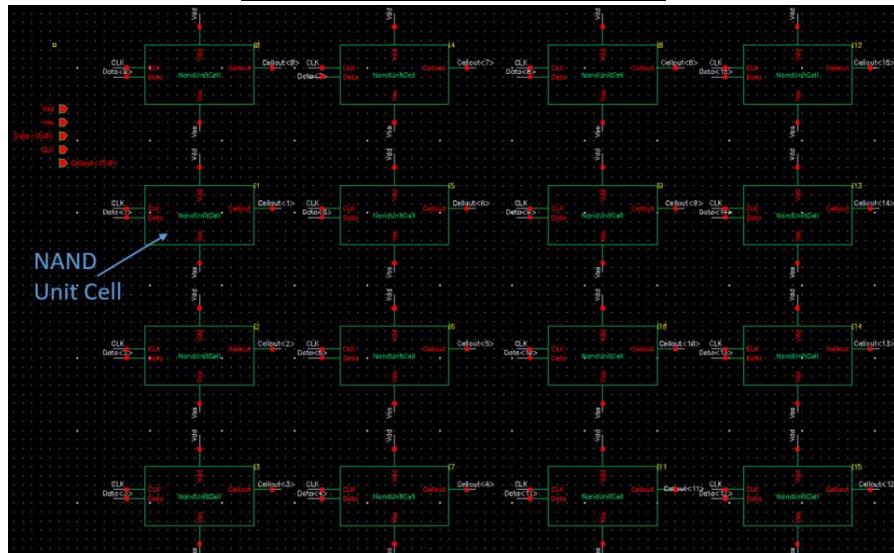
Differential Array schematic

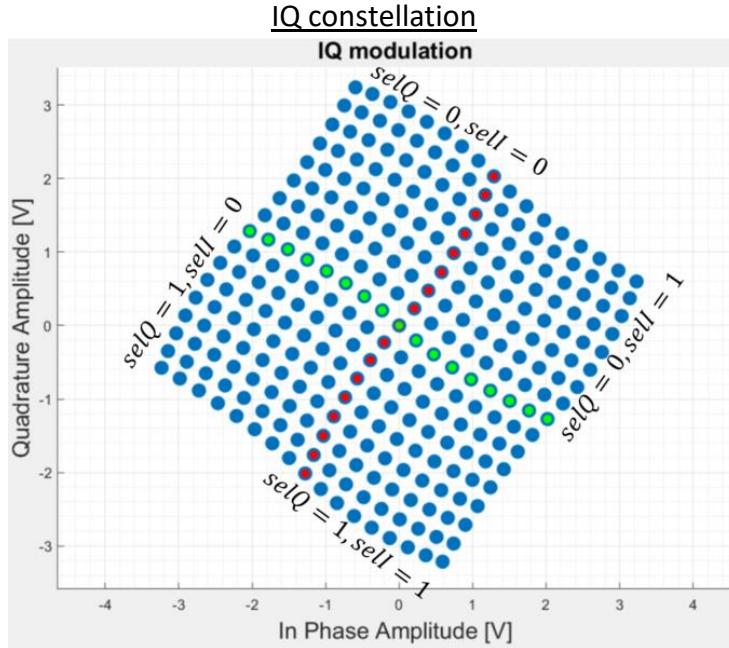


NAND Unit Cell Array schematic



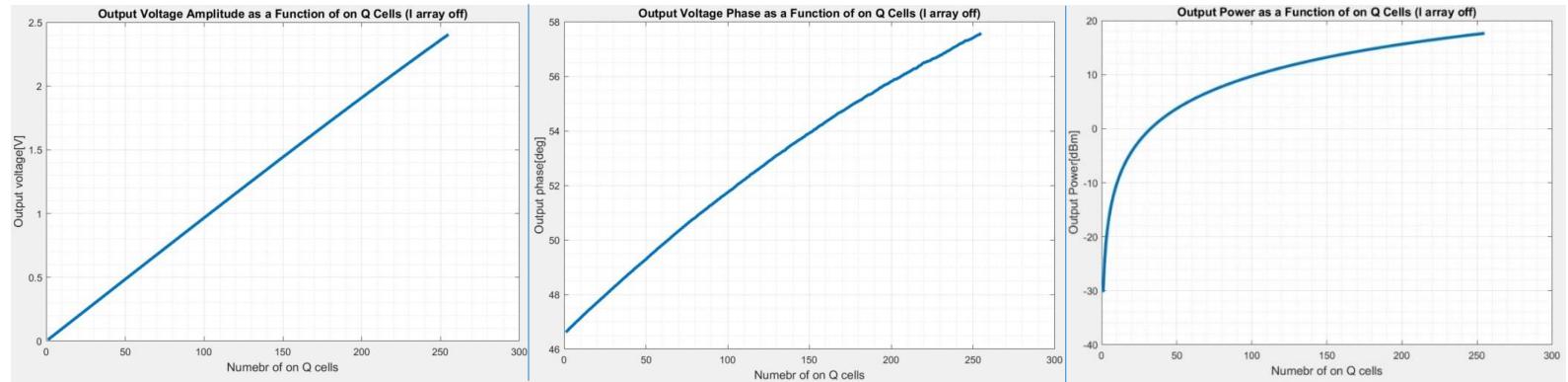
NAND Unit Cell Group schematic





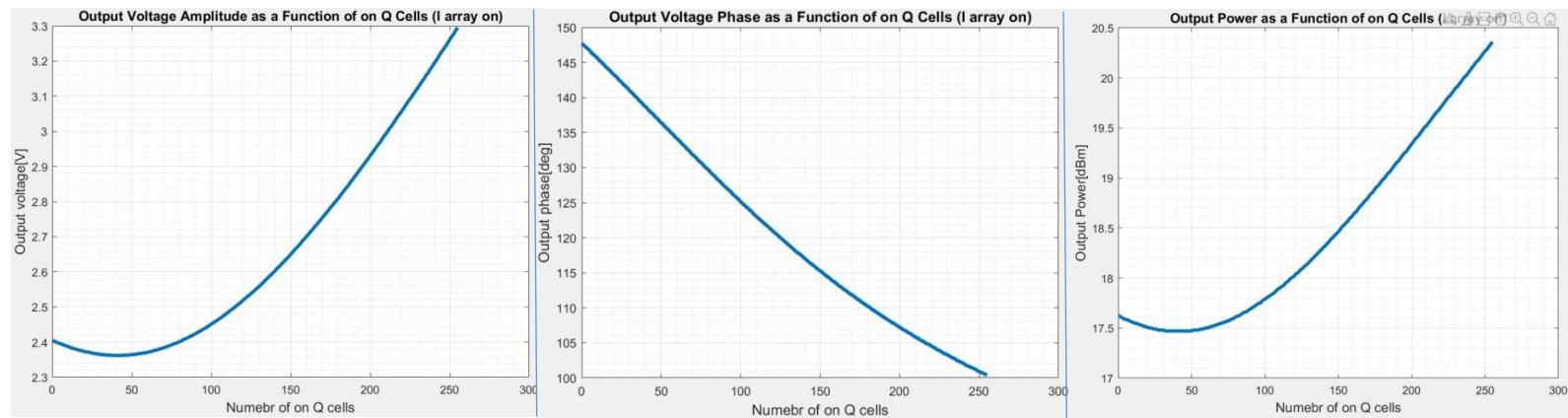
Using the Clock Control, we were able to easily transmit in all quarters of the IQ constellation plane. Next to each quarter, there are the selection bits relative to it. As before, the constellation is curvature because if the nonlinear behavior of the circuit.

Amplitude, Phase and Power when the I channel is off



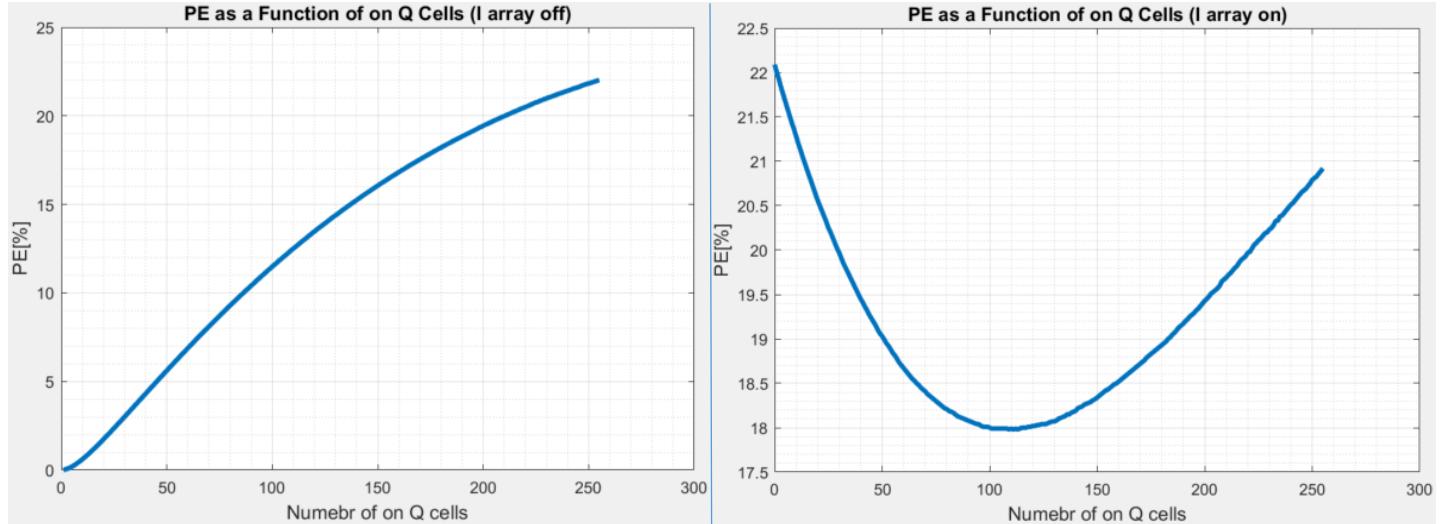
The amplitude graph looks about right. It is linear and begins at 0V. The phase graph isn't constant, as in the ideal case, due to the circuit's non-linear behavior. The power graph looks logarithmic, as it should.

Amplitude, Phase and Power when the I channel is full on



Both the amplitude and power graphs become relatively straight at some point. They are a little saturated at the beginning, but they decrease at the start. This is because of the nonlinear behavior of the circuit. The phase graph looks a little curved, as it should. However, because the simulation is done for $selQ = 0$ and $selI = 0$, as the Q channel turns on, the phase actually drops.

Power efficiency graphs

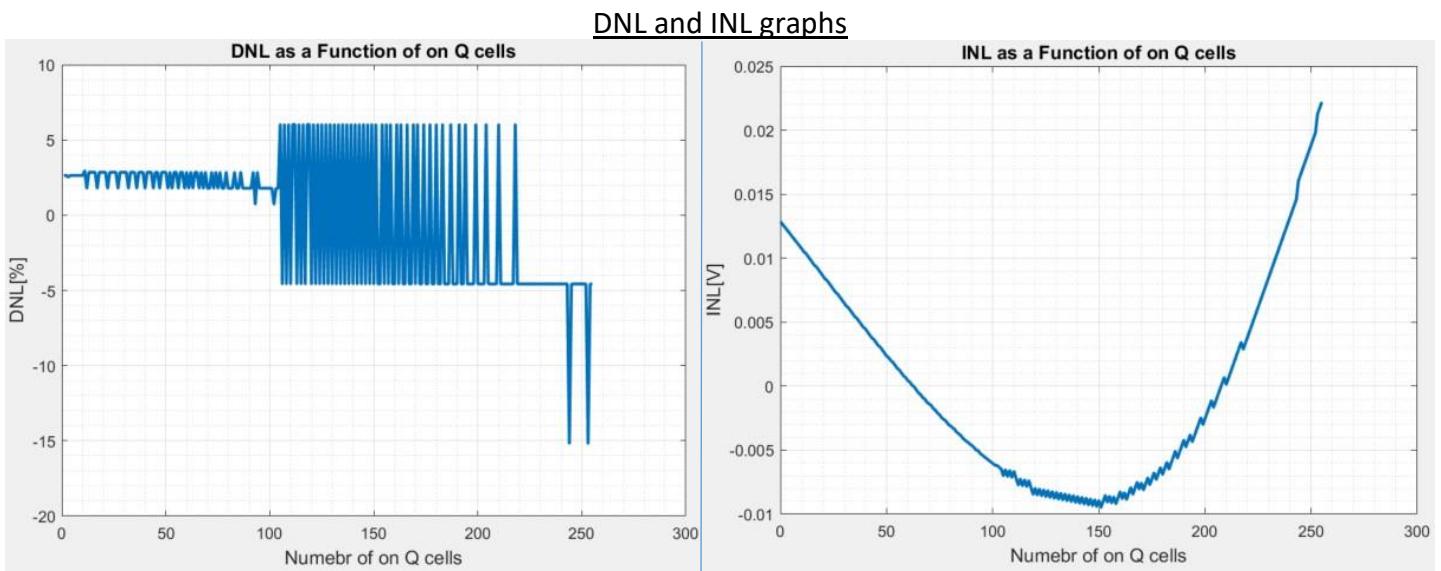


When the I array is off and the Q cells are turning on, both the power consumption and the output power increase. The output power increases enough so their ratio increases as well and we see the PE graph increases.

When the I array is on, turning on Q cells adds less to the output power, but the power consumption increases by the same amount. Thus, their ratio decreases and we see the PE graph decreases. However, at some point turning more cells on adds little to the power consumption and the PE graph increases again.

The QSCPA is made of a DAC, whose performance can be characterized by DNL and INL graphs. Differential Nonlinearity (DNL) characterizes a DAC's precision. Ideal DAC's output is linear in the input code. Ideally increasing the input code by one should increase the output by a constant step called the least significant bit (LSB). DNL plots the deviation of the step $DNL = \frac{\text{Actual step size} - \text{LSB}}{\text{LSB}} \cdot 100\%$. We take LSB as the average step size. Integral Nonlinearity (INL) is another measure of the DAC's precision. INL plots the deviation of the output from a fit linear curve $INL = \text{Actual output} - \text{Linear fit of the output}$.

The output amplitude is linear only when one channel is open, otherwise the output – input relation is $\sqrt{n_Q^2 + n_I^2}$. Thus, we plot the DNL and INL graphs when the I channel is off and the Q channel's code varies.



The LSB is $9.4mV$ and the average DNL and INL is zero. The maximum DNL is 15.17% and the maximum INL is $22mV$.

Matlab Code for extracting DNL and INL

```

coefficients1 = polyfit(Qcode(1:256),
Vout_h1mag(1:256), 1);
linearRegression1 = polyval(coefficients1,
Qcode(1:256));
INL1 = linearRegression1 - Vout_h1mag(1:256);
figure
p4 = plot(Qcode(1:256), INL1);
title('INL as a Function of on Q cells',
'FontSize', 12);
xlabel('Numebr of on Q cells', 'FontSize', 12);
ylabel('INL[V]', 'FontSize', 12);
grid on
grid minor
p4.LineWidth = 2;
fprintf('INL max is %f\n', max(abs(INL1)));
fprintf('The average INL is %f\n',
mean(INL1(1:256)));

```

```

transitionHight1 = Vout_h1mag(2:256) -
Vout_h1mag(1:255);
LSB1 = mean(transitionHight1);
fprintf('The average step hight (LSB) is %f\n', LSB1);
DNL1 = (transitionHight1 - LSB1) * 100 / LSB1;

figure
p5 = plot(Qcode(2:256), DNL1);
title('DNL as a Function of on Q cells', 'FontSize',
12);
xlabel('Numebr of on Q cells', 'FontSize', 12);
ylabel('DNL[%]', 'FontSize', 12);
grid on
grid minor
p5.LineWidth = 2;

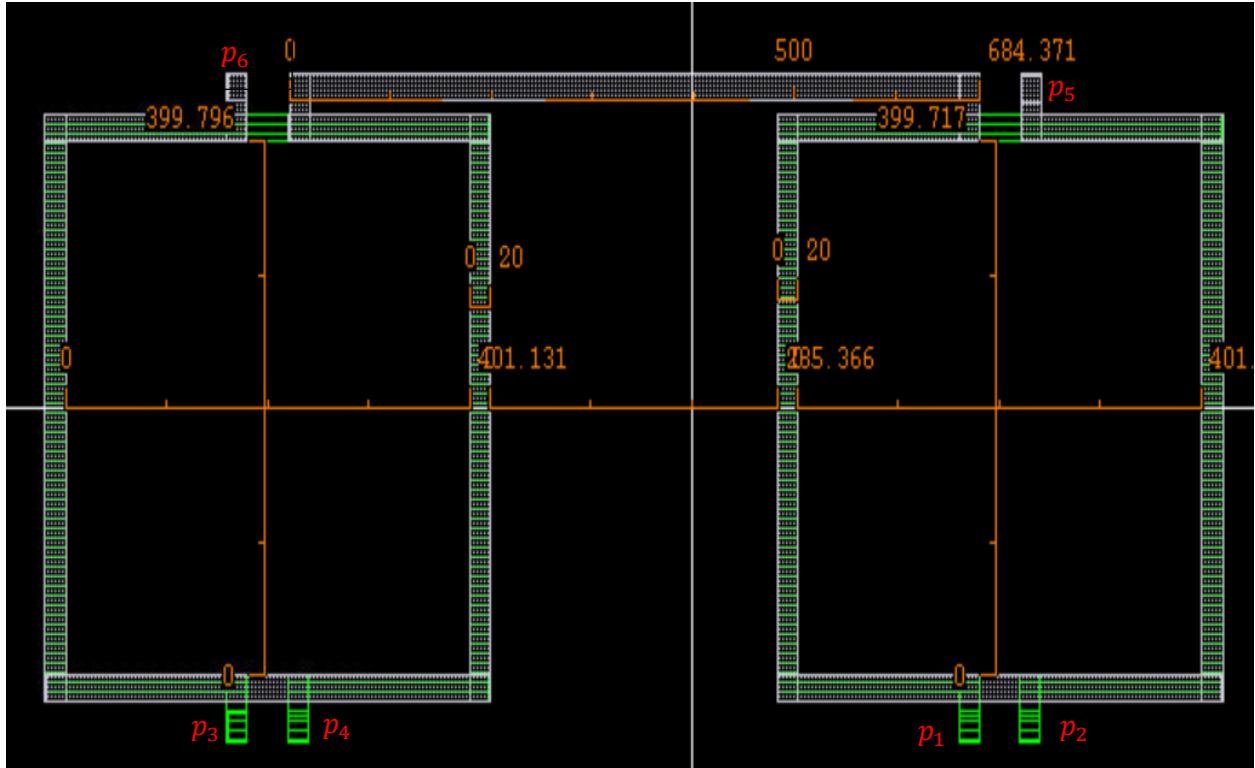
fprintf('DNL max is %f\n', max(abs(DNL1)));
fprintf('The average DNL is %f\n', mean(DNL1));

```

Momentum

For more realistic simulation and results, the series non-ideal transformers, which make up the matching network, are designed in Momentum. Momentum lets us design the physical layout of the matching network. Using the extracted physical and parasitic behavior of the layout, we get more realistic results.

Matching Network Layout



The following is the designed matching network layout. It is a 2D view with important length measurements taken by the ruler tool provided by Virtuoso layout software. The matching network comprises two identical transformers connected in series where each is built of two square shaped inductors (side length is roughly about 440 including the width of the wires) made of two different metal layers placed one on top of the other with overlapping areas to maximize the magnetic flux each inductor induces on the other and thus the coupling coefficient. The top one is made of metal AP and the bottom of metal 9. These are the thickest metals available and they serve the purpose of reducing the Ohmic resistance of each inductor, although the bottom inductors might exhibit a significantly lower resistance at the operating frequency since metal 9 is considerably thicker than metal AP and this accounts for the major difference between the resultant Q factor of the top and bottom inductors. Moreover, the inductors constituting the secondary windings might have slightly higher inductance values due to the extra wire that is used as a physical interconnect (It is not needed on the primary sides). It is important to note that the input and output pins are facing opposite

directions and one should bear that in mind so as to avoid confusion while testing the matching networks to gain its physical parameters because if the pins are misconnected, the magnetic coupling coefficient would be negative meaning that the magnetic field produced by one inductor is cancelled out and weakened by the magnetic field produced by the one parallel to it.

Matching Network Layout-3D view

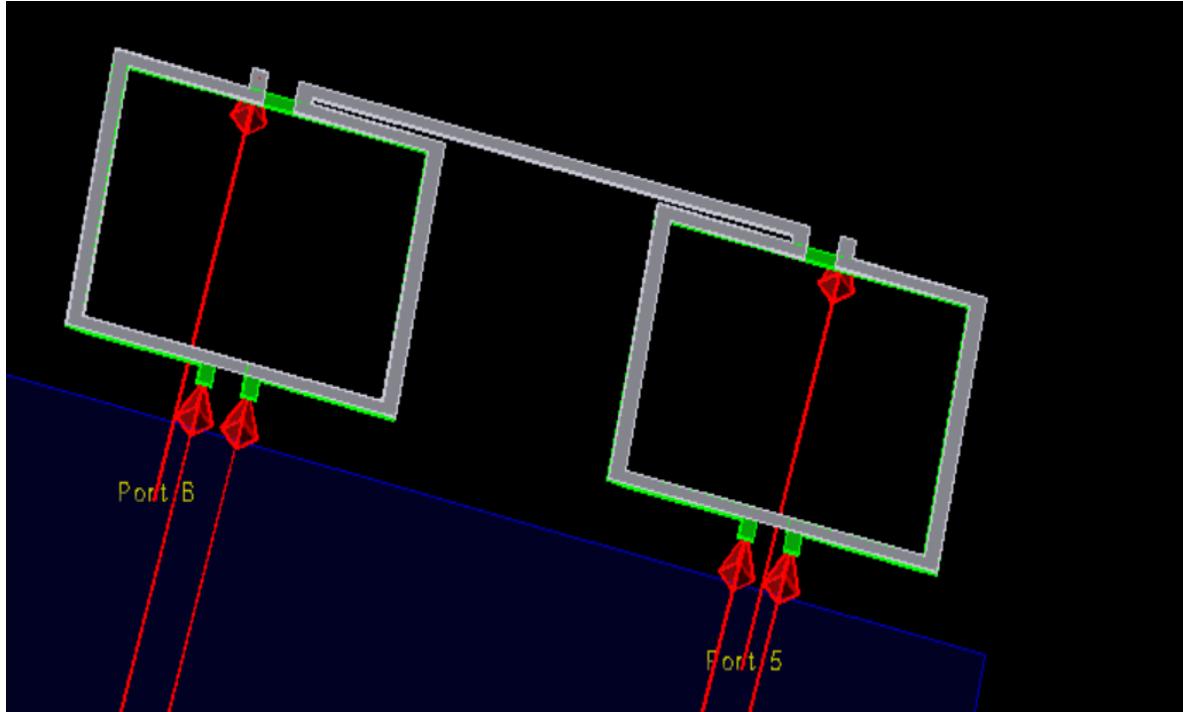
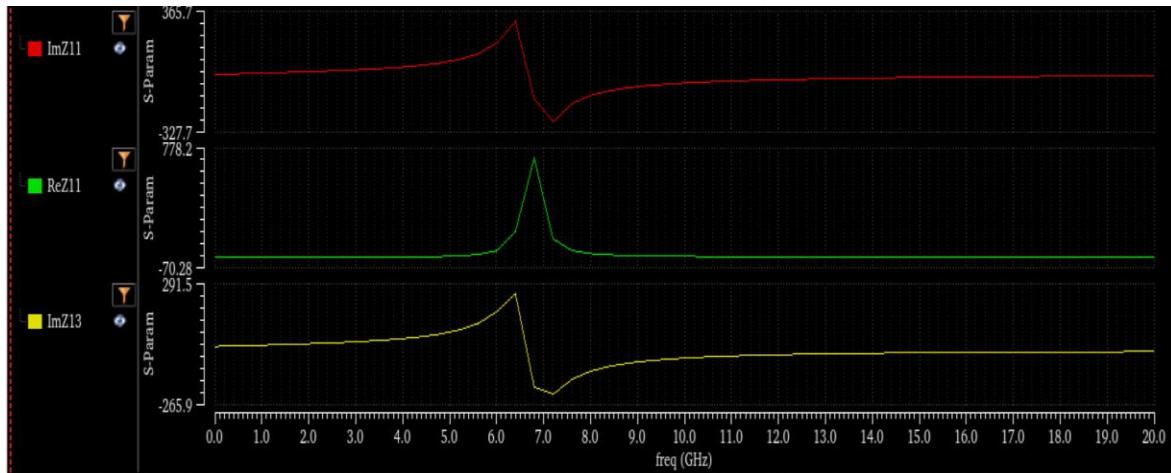


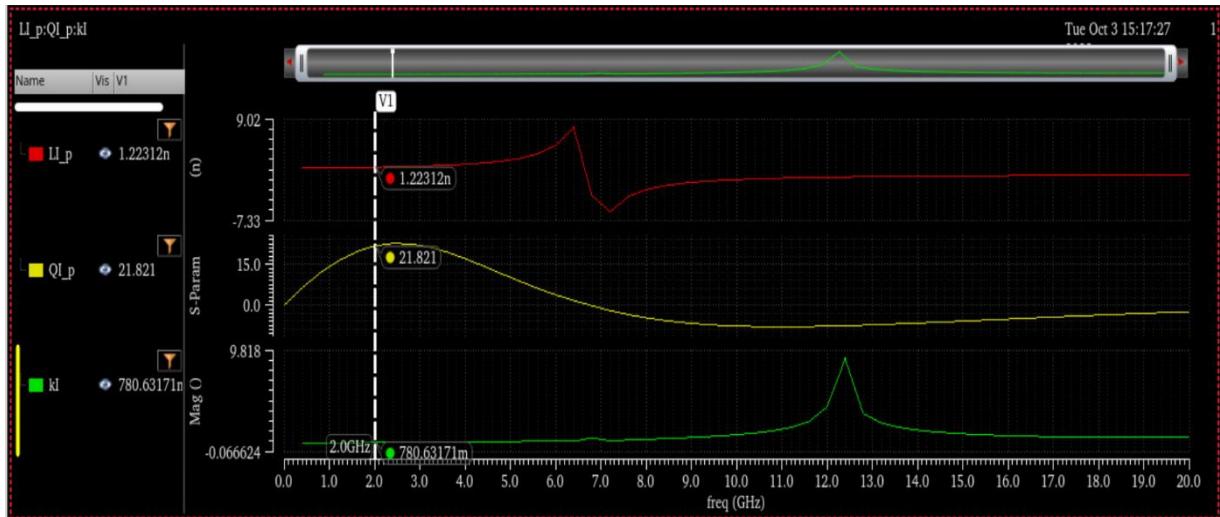
Figure 1 A 3D view of the matching network.

Parameter	Estimated value
d_{out}	$400\mu [m]$
d_{in}	0 [m] (irrelevant for a network with square shaped inductors)
W – Line width	$20 \mu [m]$
S – Line spacing	0 [m] (irrelevant for a network with square shaped inductors)
Turns ratio	Approximately 1:1

Z parameters of the I array's primary winding



Physical parameters estimation of the I array's primary winding



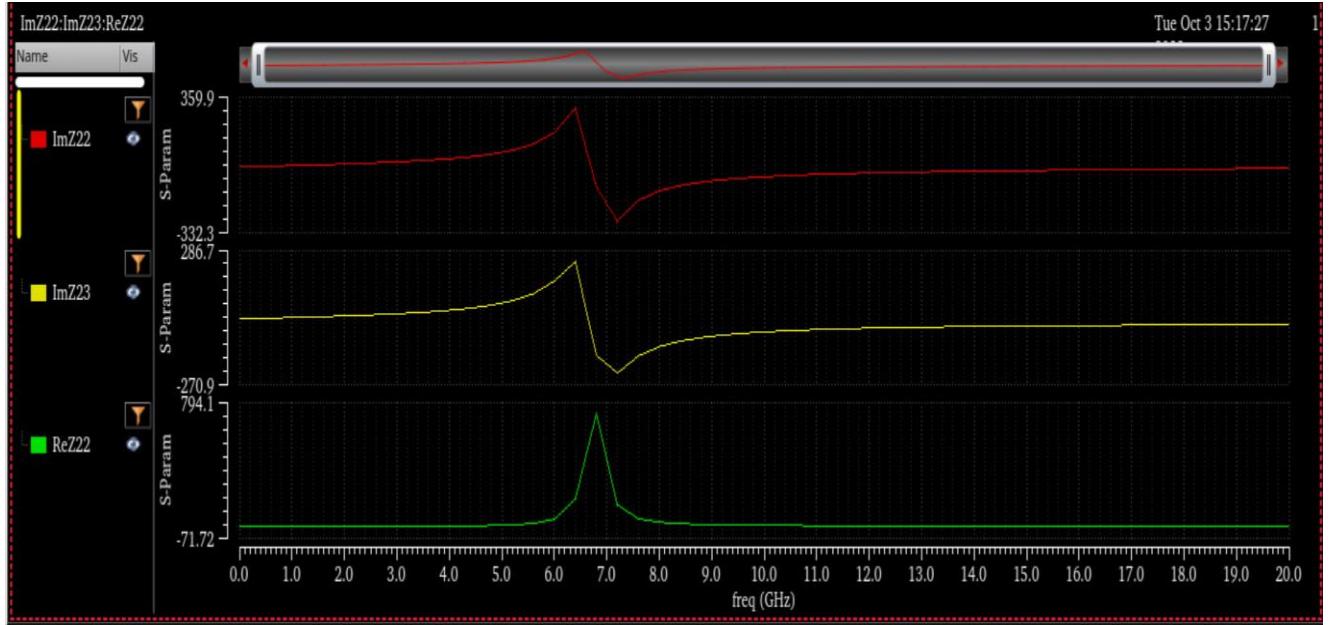
The network's Z parameters are fairly constant at the frequencies range of interest and that justifies the use of the approximation equations presented above. A conspicuous change emerges at frequency 5GHz and onwards due to the parasitics in the network that give rise to the resonance phenomena and as explained before the network components begin to unwillingly behave as capacitors instead of inductors. All the measurements were taken at 2GHz frequency.

$$L_{I(primary)} = 1.22n[\text{H}]$$

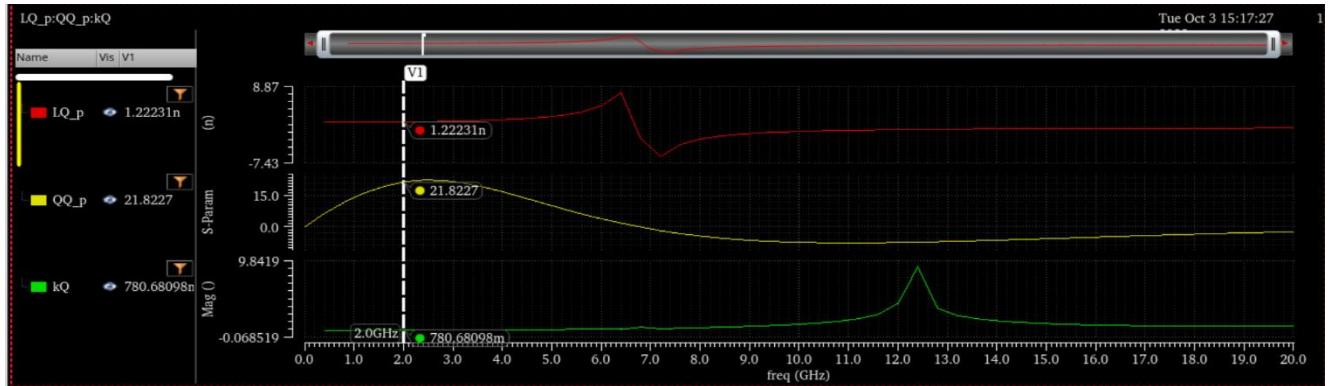
$$Q_{I(primary)} \approx 21.821$$

$$K_I = 0.78$$

Z parameters of the Q array's primary winding



Physical parameters estimation of the Q array's primary winding

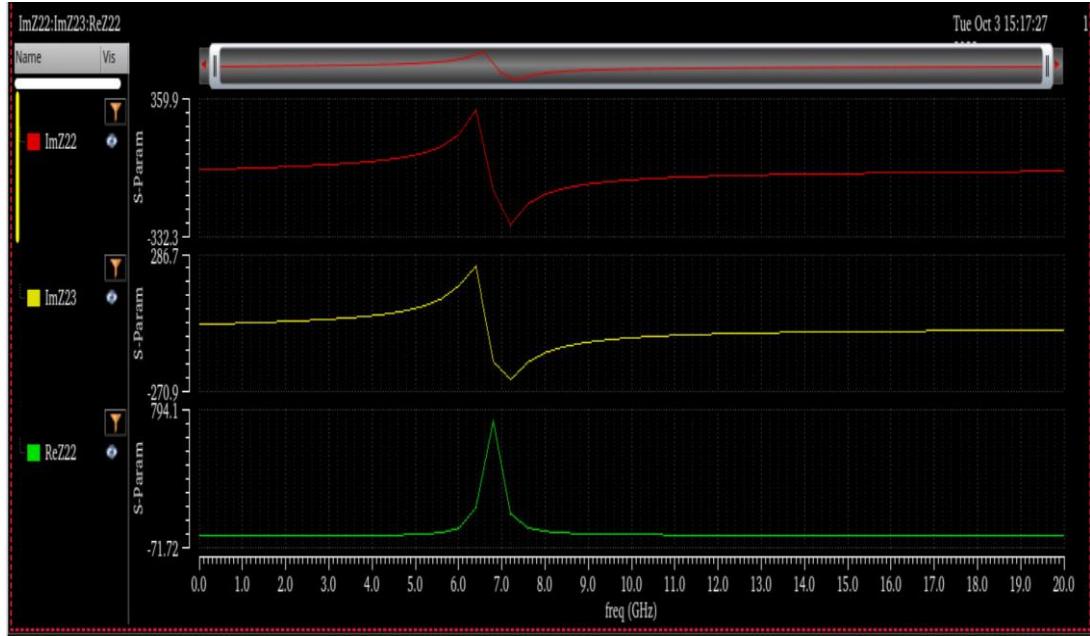


$$L_{Q(\text{primary})} = 1.22n[H]$$

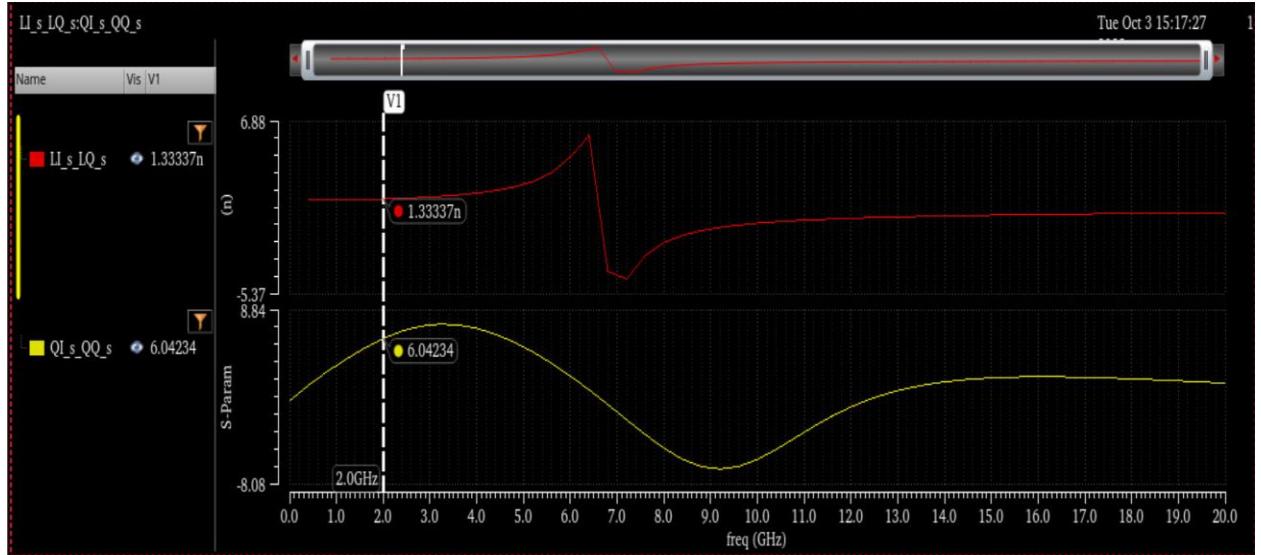
$$Q_{Q(\text{primary})} = 21.822$$

$$K_Q = 0.78$$

Secondary windings' Z parameters



Physical parameters estimation of the secondary windings



As expected, the inductance values of the secondary inductors are higher than those of the primary ones and their Q factor is about three times lower than the primary ones'.

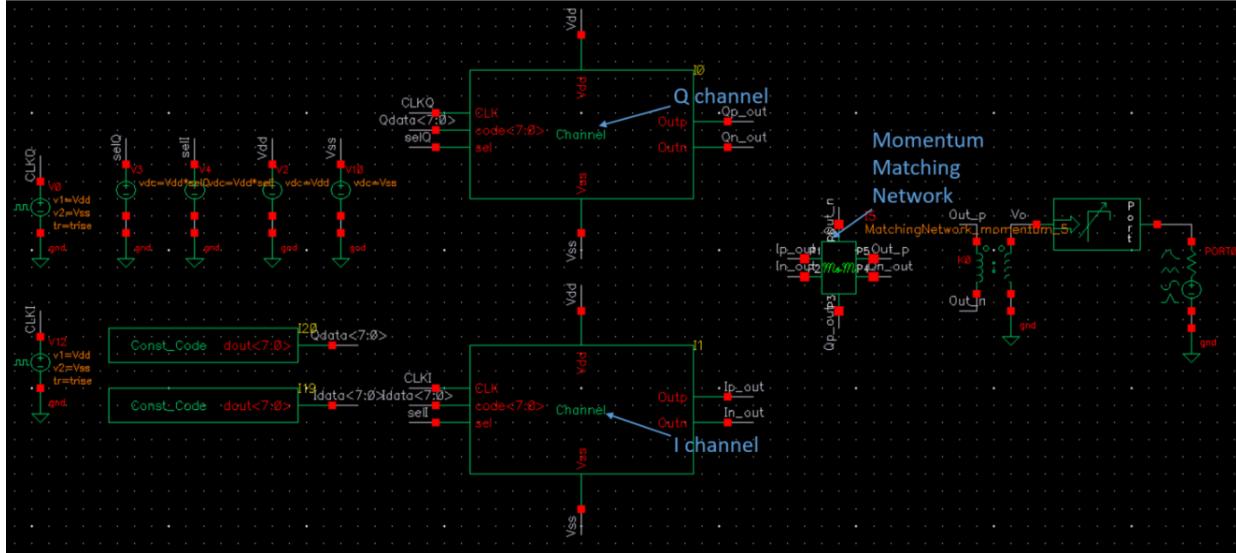
$$L_{I(\text{secondary})} \approx L_{Q(\text{secondary})} \\ \approx 1.33n[H]$$

$$Q_{I(\text{secondary})} \approx Q_{Q(\text{secondary})} \approx 6.04$$

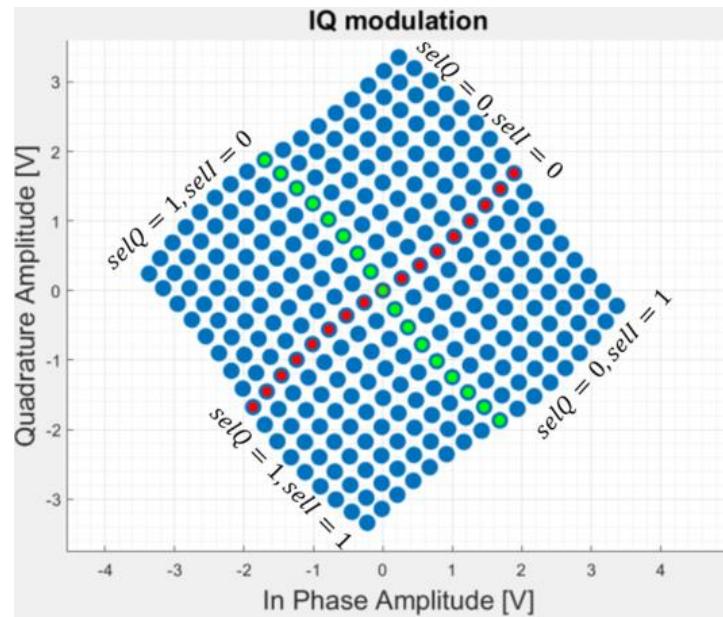
Q-SCPA with Momentum Matching network

Using the Momentum matching network we simulate and Q-SCPA again. The Momentum transformers' parameters are a little different and so the results.

QSCPA schematic

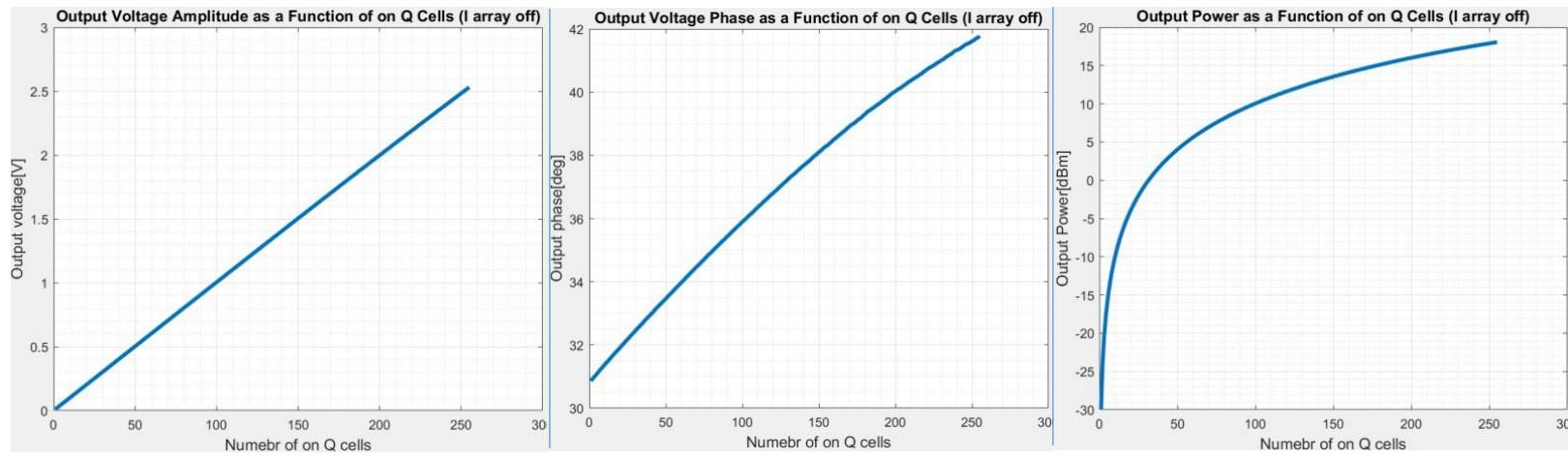


IQ constellation



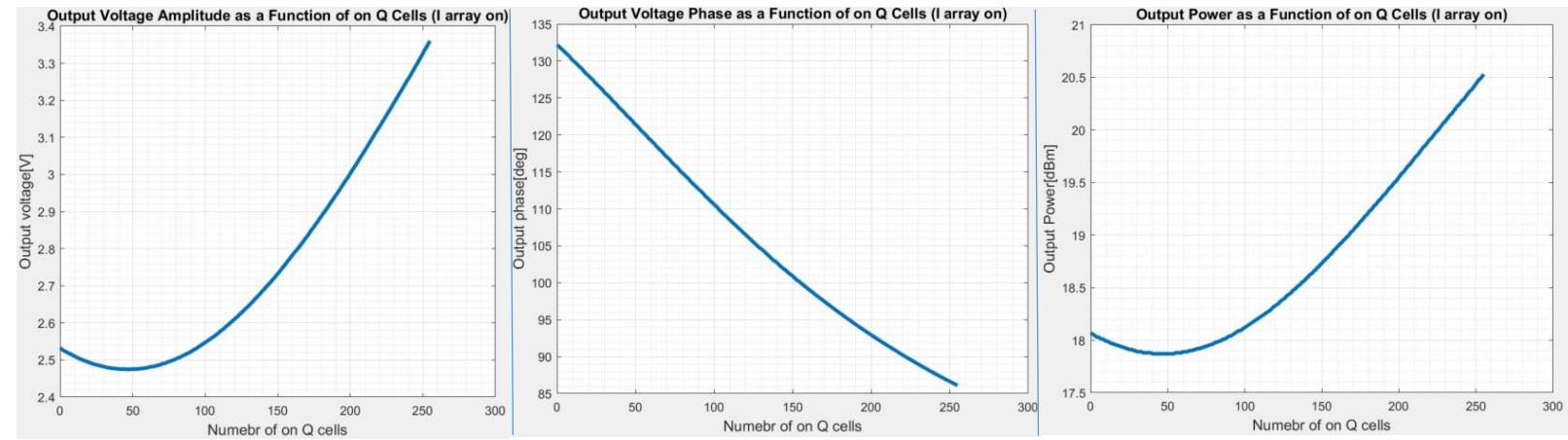
Using the Clock Control, we were able to easily transmit in all quarters of the IQ constellation plane. Next to each quarter, there are the selection bits relative to it. As before, the constellation is curvature because of the nonlinear behavior of the circuit.

Amplitude, Phase and Power when the I channel is off



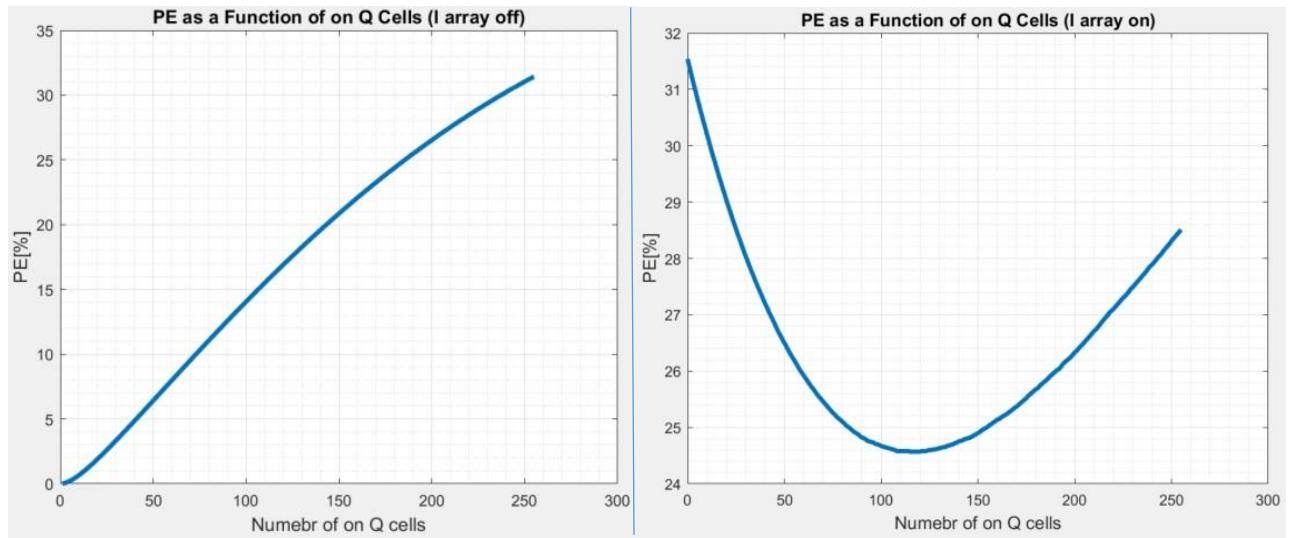
The amplitude graph looks about right. It is linear and begins at 0V. The phase graph isn't constant, as in the ideal case, due to the circuit's non-linear behavior. The power graph looks logarithmic, as it should.

Amplitude, Phase and Power when the I channel is full on



Both the amplitude and power graphs become relatively straight at some point. They are a little saturated at the beginning, but they decrease at the start. This is because of the nonlinear behavior of the circuit. The phase graph looks a little curved, as it should. However, because the simulation is done for $selQ = 0$ and $selI = 0$, as the Q channel turns on, the phase actually drops.

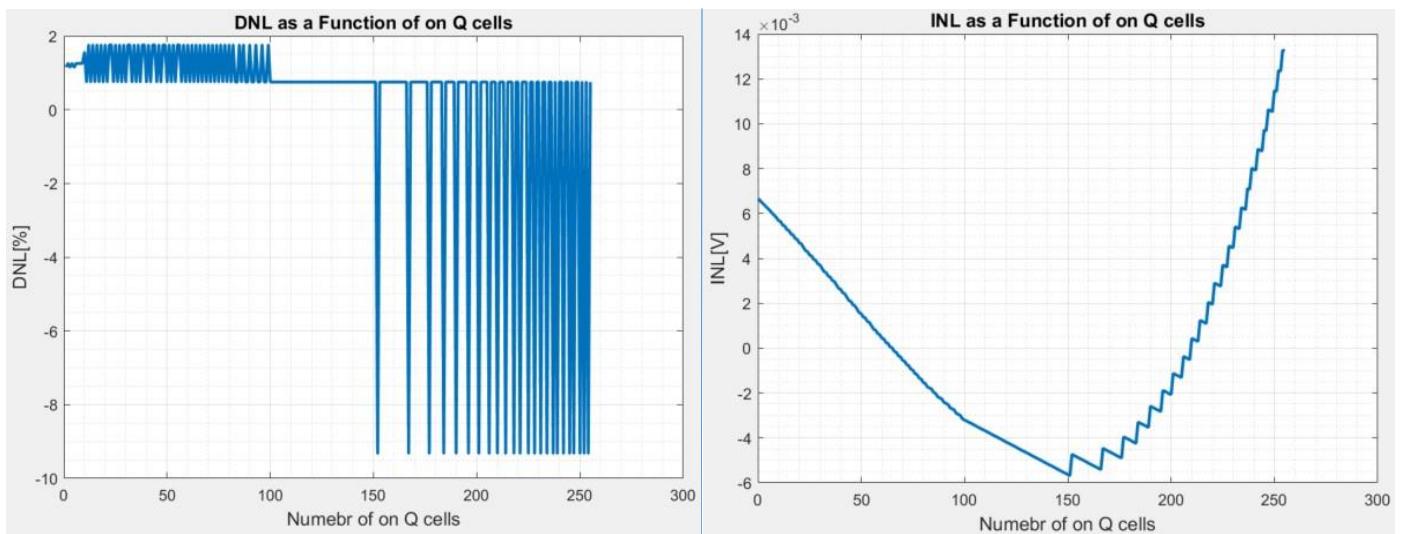
Power efficiency graphs



When the I array is off and the Q cells are turning on, both the power consumption and the output power increase. The output power increases enough so their ratio increases as well and we see the PE graph increases.

When the I array is on, turning on Q cells adds less to the output power, but the power consumption increases by the same amount. Thus, their ratio decreases and we see the PE graph decreases. However, at some turning more cells on adds little to the power consumption and the PE graph increases again.

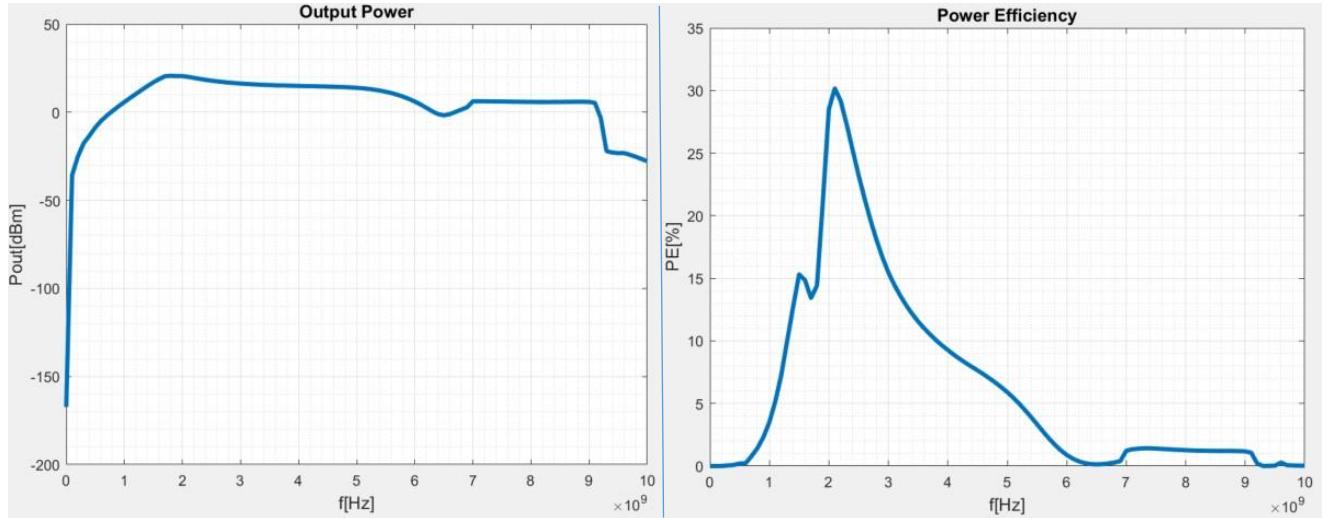
DNL and INL graphs



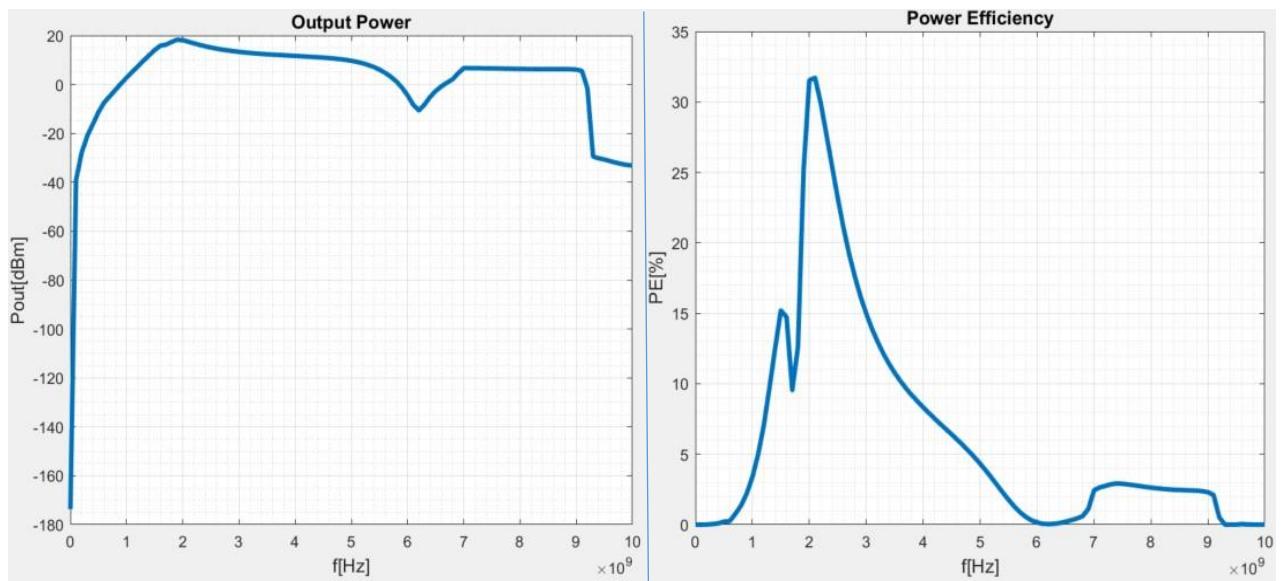
The LSB is $9.9mV$ and the average DNL and INL is zero. The maximum DNL is 9.3% and the maximum INL is $13.2mV$.

Our design is made for 2GHz transmission. We can see how it behaves for other transmission frequencies by changing the clock frequency. We look at the output power of the fundamental frequency and the PE of the circuit.

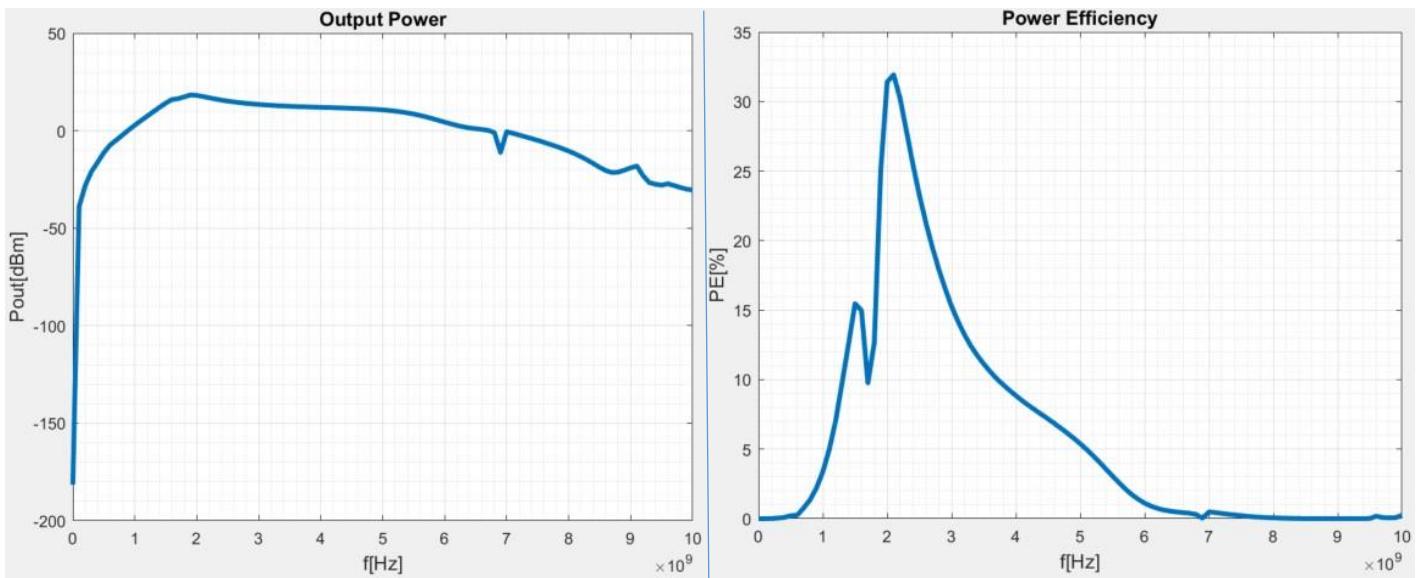
Both I and Q arrays are fully on



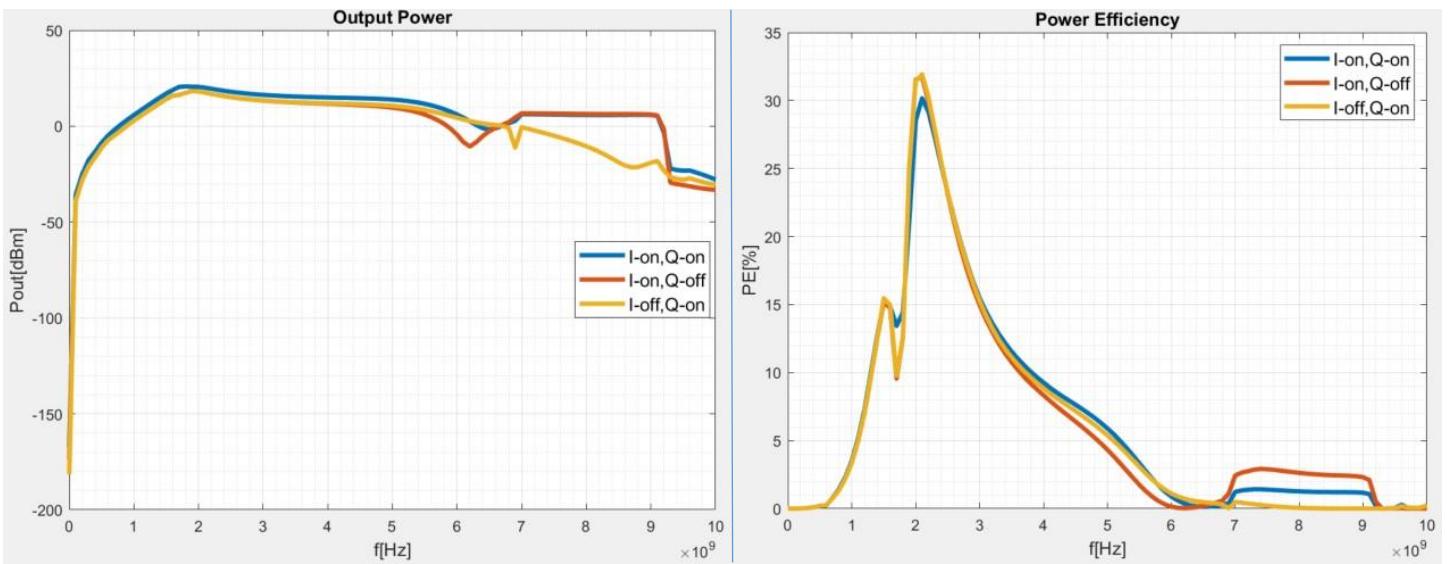
I array is fully on and Q array is off



I array is off and Q array is fully on



Combined results



Around 2GHz the output power is larger when both arrays are on, but the PE is larger when only one channel is on. It fits the graphs of the output power and PE as a function of on cells. We get peak output power before 2GHz. It makes sense, because we designed the matching network so the output power, when both arrays are on, is above 20dBm and not for maximum output power. Thus, the matching network achieves maximum output power for different frequency. As the frequency is further from 2GHz the matching network is less effective, so less power is transferred to the load and the output power and PE drop.

At high frequencies, the behavior of the circuit's transistors and the matching network's transformers change. For example, the transistors' Gate current increases and the transformers' capacitance is more dominant. These might be the cause of the not smooth behavior of the graphs, beyond 2GHz frequencies.

Matlab code for extracting the frequency graphs

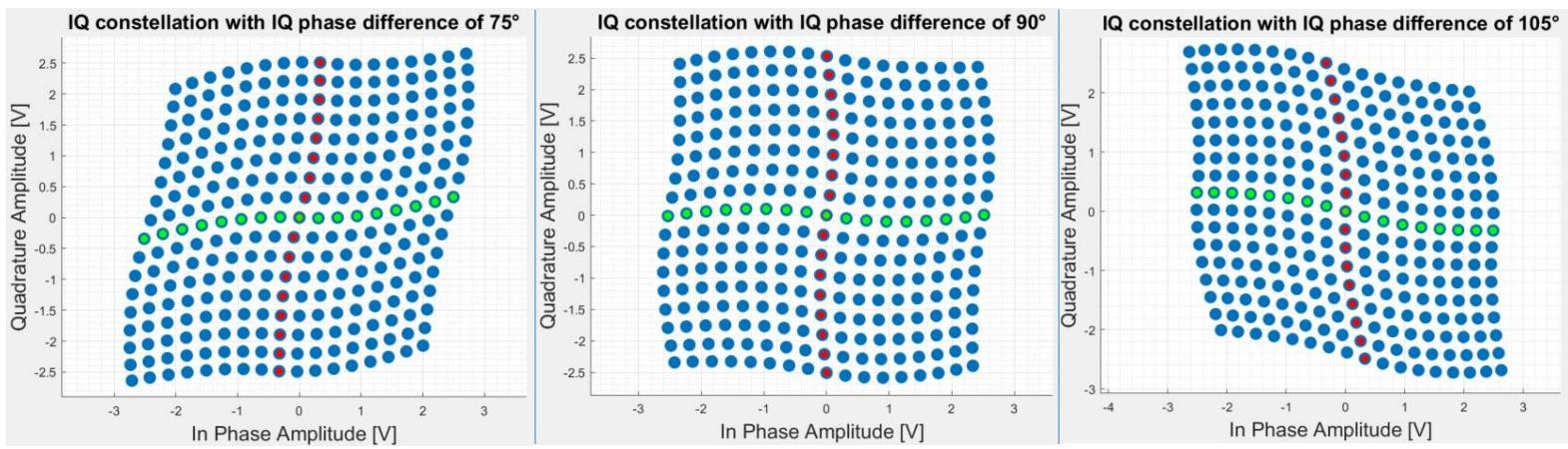
<pre>% Output power simulation p1 = plot(frf(1:101), PoutIQ(1:101)); hold on p2 = plot(frf(1:101), PoutI(1:101)); p3 = plot(frf(1:101), PoutQ(1:101)); title('Output Power', 'FontSize', 12); xlabel('f[Hz]', 'FontSize', 12); ylabel('Pout[dBm]', 'FontSize', 12); grid on grid minor lgd = legend('I-on,Q-on', 'I-on,Q-off', 'I-off,Q-on'); lgd.FontSize = 12; p1.LineWidth = 3; p2.LineWidth = 3; p3.LineWidth = 3; hold off</pre>	<pre>% Power efficiency simulation figure p1 = plot(frf(1:101), PEIQ(1:101)); hold on p2 = plot(frf(1:101), PEI(1:101)); p3 = plot(frf(1:101), PEQ(1:101)); title('Power Efficiency', 'FontSize', 12); xlabel('f[Hz]', 'FontSize', 12); ylabel('PE[%]', 'FontSize', 12); grid on grid minor lgd = legend('I-on,Q-on', 'I-on,Q-off', 'I-off,Q-on'); lgd.FontSize = 12; p1.LineWidth = 3; p2.LineWidth = 3; p3.LineWidth = 3; hold off</pre>
---	--

IQ imbalance

When using IQ modulation we wish that the I and Q components will have exactly 90° difference between them. Additionally, we wish the gain of the I and Q channels to be the same. However, in reality the phase is not exactly 90° and the gain is not the same between the channels. As a result, the IQ constellation becomes distorted.

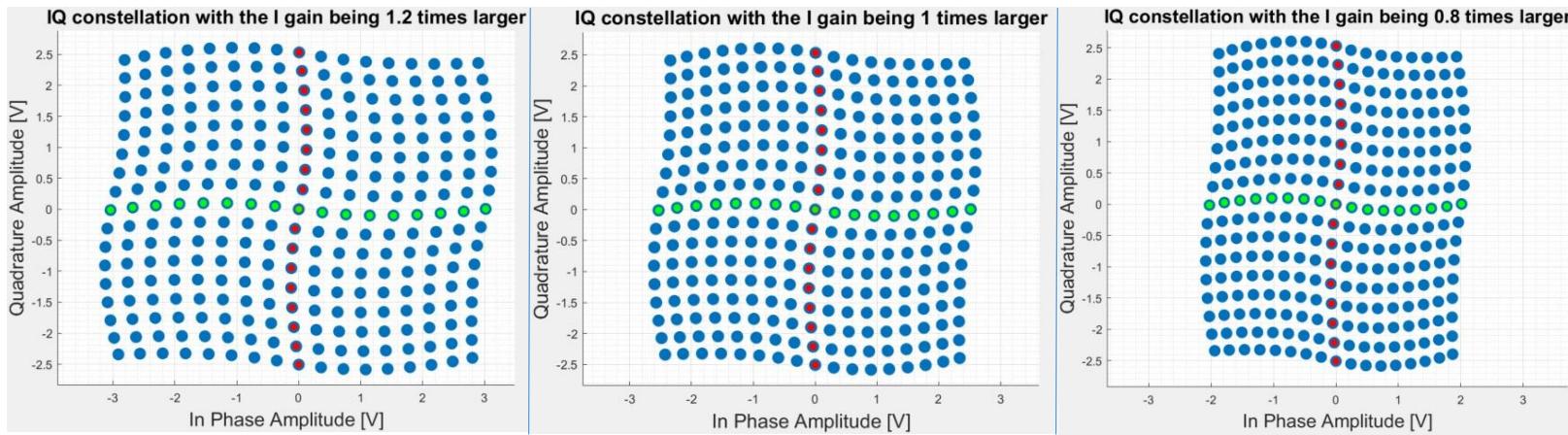
For simplicity, we cancel the constellation phase offset shown previously. Additionally, we introduce large imbalances, to see their effect clearly. In reality, we expect them to be relatively small.

IQ constellation with phase imbalance



The red and green points refer to cases where the I and Q arrays (channels) are off, respectively. The phase imbalance causes the I and Q components to not be orthogonal. We see that by looking at the phase between the red and green dots. The constellation is stretched across one diagonal and shrinks across the other, taking the shape of a rhombus.

IQ constellation with gain imbalance



The red and green points refer to cases where the I and Q arrays (channels) are off, respectively. The gain imbalance causes the I and Q ranges to be different. When the I gain is larger the constellation is stretched horizontally. When the gain is smaller, the constellation is stretched orthogonally.