ECE431 Final Project
Vector Modulator

Jay Williams, Azra Rangwala, Ilona Lameka, Ridwan Hussain

December 10 2024

1 Introduction

1.1 Specifications

The vector modulator was designed for 16 QAM (Quadrature Amplitude Modulation), with a carrier frequency of 5.4 GHz and a minimum double sideband bandwidth of 1.8 GHz. The symbol rate is 900 MHz, with 4 bits per symbol. Two of the bits were to be used for a 0/180° phase modulation while the other 2 bits were for attenuation or no attenuation. The modulator must have a maximum of 8.5 dB insertion loss, along with a 1.4:1 maximum VSWR.

The constellation must have a maximum of ± 3.5° phase accuracy and ± 0.7 dB amplitude variation over the entire bandwidth. The modulator parameters must be achieved without use of amplifiers (a purely passive design) and transmission lines must use the Rogers 4003 substrate material.

1.2 Assumptions

The design was started with some assumptions about the circuit and its signals.

There were four bits used in total to control the phase and attenuation for the constellation, which was represented by a ±2.5 VDC. A +2.5V voltage source represented a bit value of 1, while a -2.5V value was used for a bit value of 0.

The symbol rate of 900 MHz was taken into account when searching for a mixer that would function at that rate. Since most mixers do not support a wide bandwidth for their RF/LO ports, it was determined that the bits would be sent through the IF port. The assumption was that most mixers have an IF port that supports DC - low GHz frequencies.

Additionally, PIN diodes generally cannot switch faster than 20 nanoseconds, and with our requirement of around a 1 nanosecond switch, we assumed that the PIN diodes act as ideal switches, and no transient analysis was done.

2 Vector Modulator Architecture

While the overall design followed the recommended block diagram, multiple options were available for the four blocks in the design. Figure 1 is a block diagram that describes the high level overview of the vector modulator. Figure 2 displays what the ideal constellation output would look like with the coding pattern chosen for this project. Gray Coding was implemented with the current coding scheme, which means that all adjacent points are only different by a single bit. This helps to minimize overall bit error rate since a larger distance error must be achieved to switch two or more bits.

The first stage of the modulator is a quadrature divider that splits an RF input into two signals with a 90° phase difference between them. One of the inputs would have no phase shift with reference to the RF input, while the other would be coupled to a 90° phase shift. Both output signals would ideally be equal in magnitude.

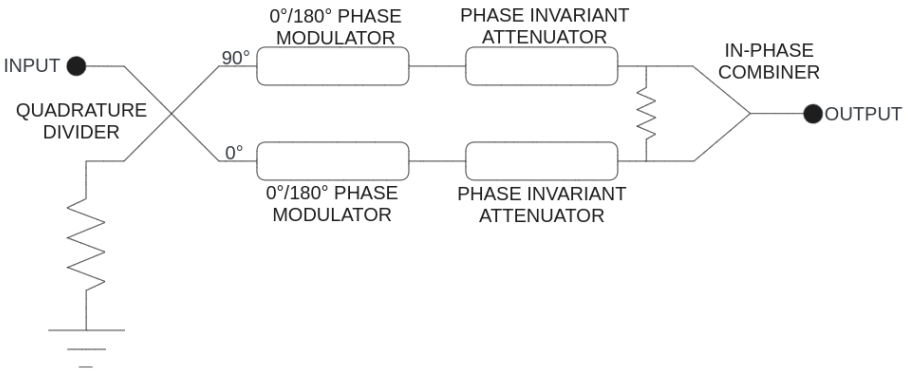

Figure 1: Recommended Block Diagram

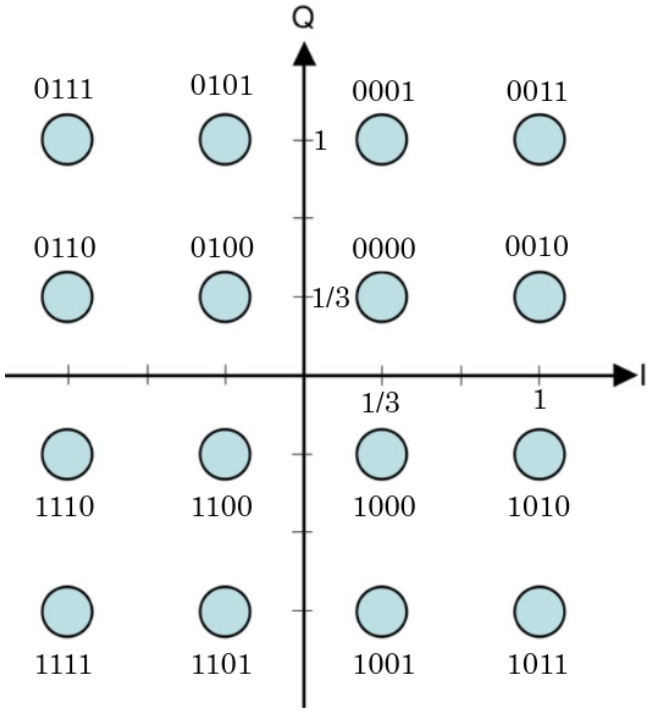

Figure 2: Final Ideal Constellation

Multiple possible quadrature dividers exist, such as branch-line hybrids, coupled line couplers, and Lange couplers.

The branch-line coupler is able to produce two outputs with a 90° phase shift between them, along with 3 dB coupling for equal power division. Due to its symmetry, any of its ports can also be used as the input for the 5.4 GHz

signal. However, the branch-line coupler is limited to a bandwidth of $10\% - 20\%$. Due to the required 1.8 GHz ($\sim 33\%$) bandwidth of the vector modulator, the branch-line hybrid was eliminated as a potential part.

The coupled line coupler was another quadrature divider that was capable of creating a 90° phase shift between the outputs, however, the coupling was too loose to achieve coupling factor of 3 dB. The Lange coupler solves this issue, which has multiple parallel coupled lines for tight coupling. Since the Lange coupler can achieve 3 dB coupling, create a 90° phase shift, and works at a 1.8 GHz bandwidth, it was selected as the first stage of the vector modulator.

With the two outputs being 90° apart, the next stage was the 0°/180° modulator. This stage can be achieved with using a mixer as a modulator or phase shifters. One common phase shifter is the switched line phase shifter, however it is very limited in bandwidth. An alternative to the switched line phase shifter would be the all pass filter, which has a high and low-pass arm that each provide a phase shift. The all-pass filter is made up of lumped capacitors and inductors, which brings an increased complexity to the design. For this reason, a double-balanced mixer was selected, that can have its IF port be driven by a DC voltage representing a bit value. The bit was coded such that a "0" meant no phase shift, and a "1" meant 180° phase modulation.

The third stage is the phase invariant attenuator. The purpose of this stage is to either have the input be attenuated by a certain amount, bringing the constellation closer to the origin, or have no attenuation on the signal, leaving the constellation near the edge. Since this circuit is a phase sensitive circuit, it critical that regardless of which attenuation path the signal takes, there is very little phase variation or the phase variation is the same relative to each other. The "0" bit means that the signal will be attenuated, and the "1" bit means that the signal will not be attenuated.

For the no attenuation path, a quarter-wave transmission line was used in order to have minimal attenuation. For the attenuated path, a π-resistor network was used. In order to determine the proper attenuation value, the final constellation chart was referred. Since there are 4 points per column/row, and the outer points were normalized to zero, to get equidistant points, each point must be 2/3 away from each other relative to the outer most point. As the result, the inner points must be 1/3 of the value of the outer points, and converting that to the dB scale leads to the following calculation. $20log(1/3) = 9.5dB$ This means the π-network should provide approximately 9.5 dB.

The final stage is the in-phase combiner that sums the two signals together. The power combiner must also have minimal phase variation as mentioned before. The design used to achieve this was a Wilkinson Power combiner since it provides minimal attenuation at 3 dB, high isolation between the two input ports to prevent coupling, maintains phase coherence to reduce the phase variation from this stage, and allows for 50Ω impedance matching to minimize the reflection losses to reduce the VSWR.

Figure 3 displays what the final block diagram looks like with the various blocks described above. The Lange Coupler is used for the quadrature divider, the double-balanced mixer is used for the 0°/180° phase modulation, the π-

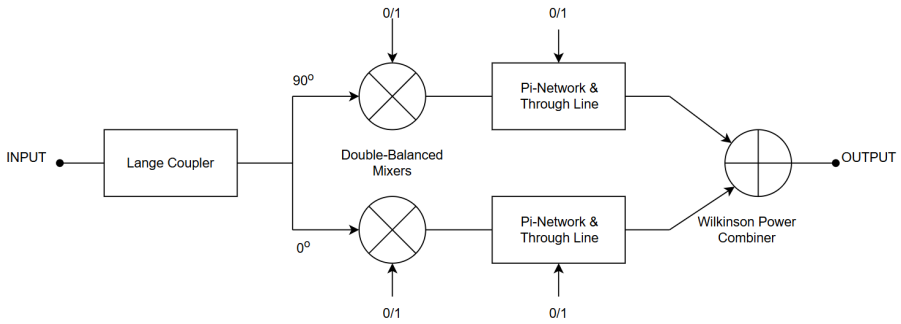

Figure 3: Final Block Diagram

network and through-line circuit was used for the phase invariant attenuator, and a Wilkinson Power Combiner was used for the in-phase combiner.

3 Block Designs

3.1 Quadrature Divider

Since the Lange coupler was selected for the quadrature divider block, a new decision arose as to how many "fingers" the Lange coupler should use. The more fingers a Lange coupler has, the better coupling it can achieve for its through and coupled ports. For this reason, the 8 finger coupler, `MLANG8`, was selected from the ADS library.

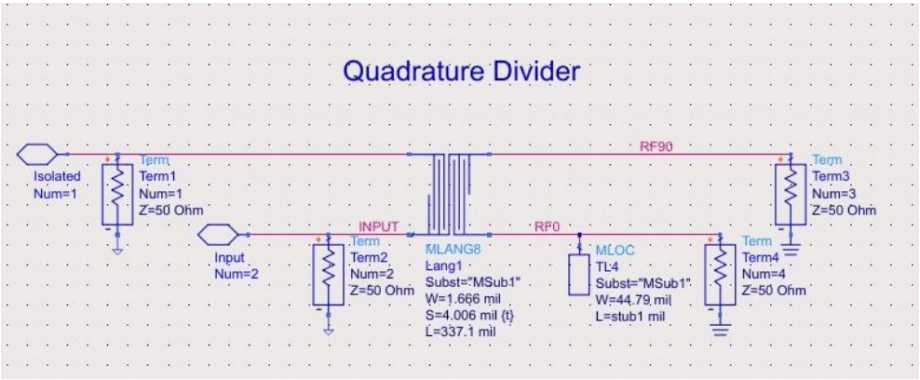

Figure 4: Lange Coupler Schematic

The Lange coupler schematic in Figure 4 shows the two outputs, with the top right port being the coupled output while the bottom right is the through. The current schematic features an open circuit stub for the through output. This stub is used for tuning purposes, since the Lange coupler in Figure 4 is

connected to the rest of the circuit.

The width, gap length, and length of the `MLANG8` were calculated with LineCalc to get the correct phase and magnitude results. In a Lange coupler, the smaller the gap between the "fingers", the stronger the coupling is for the outputs. The critically-coupled result would feature a 3dB coupling at 5.4 GHz.

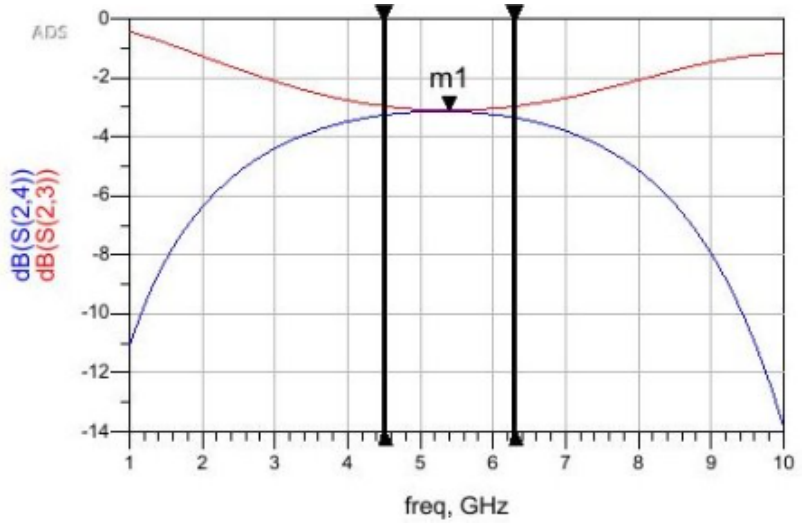

Figure 5: Lange Coupler Coupling Graph

Figure 5 shows the coupling response of the Lange coupler, with multiple measurements taken in the bandwidth. At 5.4 GHz, the magnitude of the coupling was determined to be -3.112 dB. At 4.5 GHz, the coupling variation is 0.389 dB while at 6.3 GHz it is 0.297 dB.

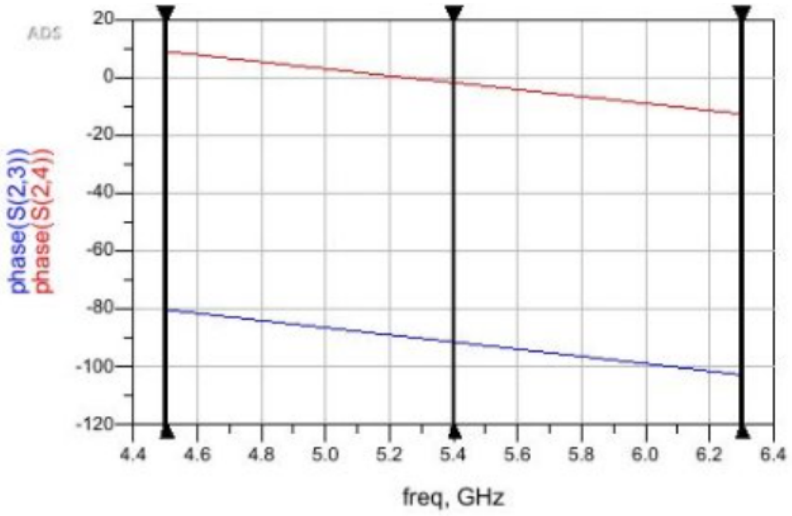

Figure 6: Lange Coupler Phase Response

Figure 6 shows the consistent phase shift between the two outputs, with the coupled port being 90° behind the through port. At the center frequency of 5.4 GHz, the phase difference was 89.643°. At 4.5 GHz the phase difference was 89.3° and at 6.3 GHz, 89.643°. The overall phase variation across the bandwidth is 1.697°.

Once the correct `MLANG8` measurements were determined, it was connected to the rest of the blocks and tuned to get the desired response. Since this is a custom-designed part, its widths and lengths were checked to make sure they can be achieved by a manufacturer.

3.2 0°/180° Modulator

The mixer for the 0°/180° modulator was established to be a double-balanced mixer. A double-balanced mixer was chosen as it has good isolation between all 3 ports, rejects all even harmonics of RF and LO, and has a higher IIP3 than a single-ended mixer. Two options were presented, creating the mixer from components such as transformers, diodes, capacitors and inductors or using a real mixer model. Since a real mixer model would have better parameters then a mixer built from scratch, the Marki library of mixers was explored. The mixer needed to have an RF/LO port that works at 5.4 GHz, and an IF port that can take data at 900 MHz. The Marki MM1-0312H[1] mixer was selected, where it has a typical conversion loss of 7 dB at IF frequencies of DC-4.5 GHz and RF/LO frequencies of 3-12 GHz. The mixer was connected so that it was used as a modulator, where the RF has a sign change based on the sign of the LO.

[1]Datasheet: https://markimicrowave.com/products/surface-mount/mixers/mm1-0312hsm-2/datasheet/

To do this, RF must go in the LO port, the LO square wave must go into the IF port, and the output must come out of the RF port, as seen in Figure 7.

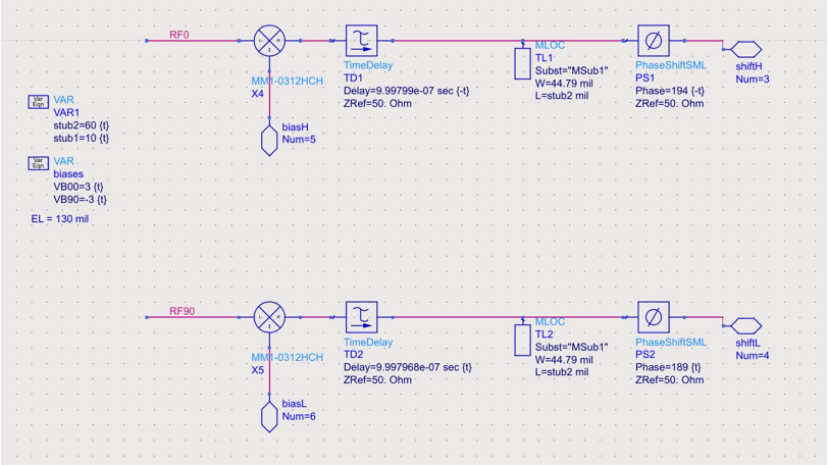

Figure 7: 0/180 Degree Modulator

An ideal Time Delay and Phase Shift block were added to the schematic so that the following S-parameter simulations would show the phase variation and amplitude variation clearly. These blocks will not be implemented in real life, as it is just to make the simulation outputs look nice.

Figures 8-11 show the S-parameters for states 00xx, 01xx, 10xx, 11xx, just for the modulator. State 00xx was used as the reference when calculating the phase variation of each of the next states.

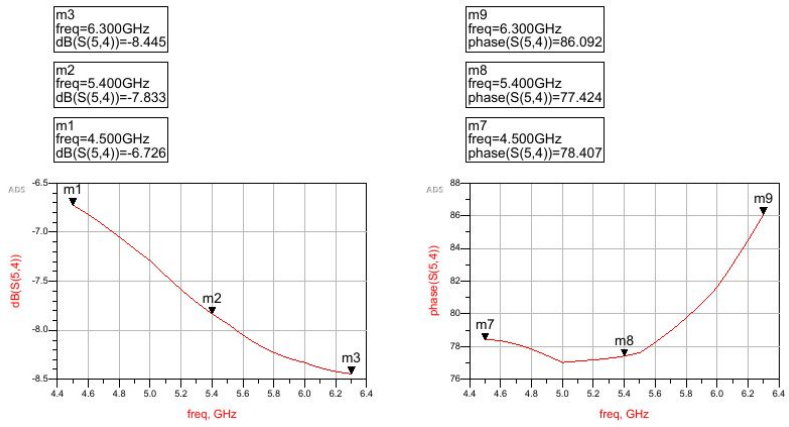

Figure 8: State: 00xx

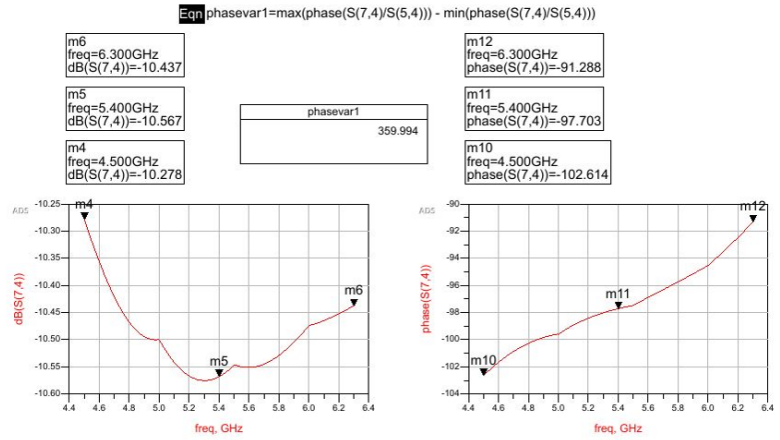

Figure 9: State: 01xx

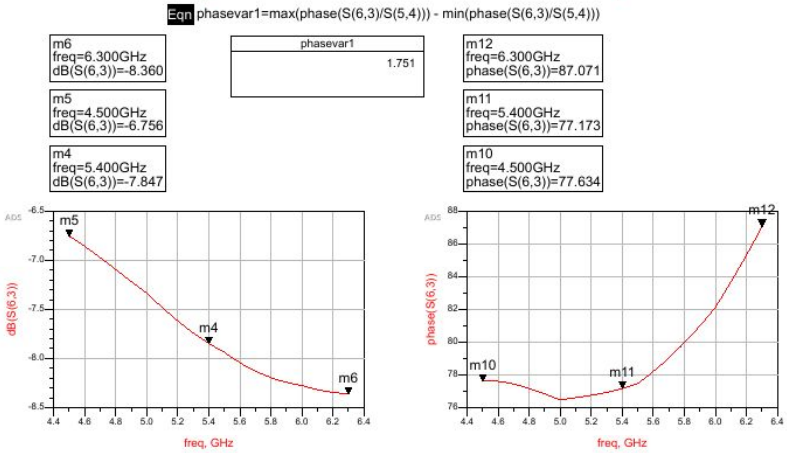

Figure 10: State: 10xx

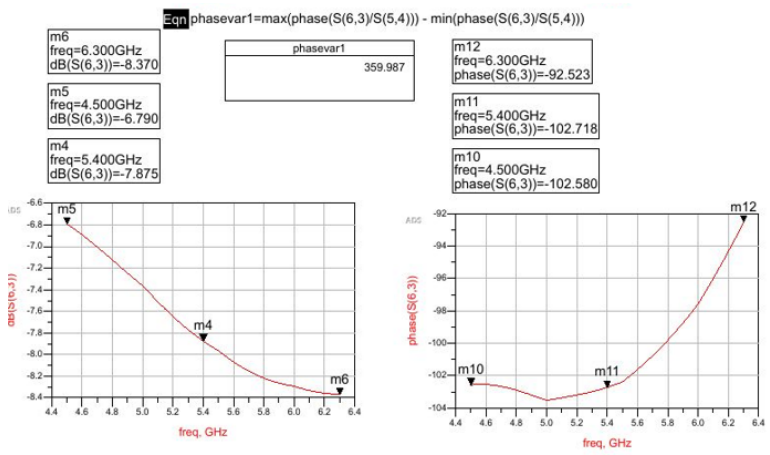

Figure 11: State: 11xx

3.3 Phase Invariant Attenuator

As mentioned previously, the phase invariant attenuator relies on a simple π-network to provide attenuation based on the bit signal. Depending on the bit, the signal would be attenuated either by 0 dB or 9.5 dB, corresponding to a normalized magnitude of 1 and 1/3, respectively. A through line with matched impedance was used for 0 dB attenuation, while a π-network provided 9.5 dB for the input signal.

To determine if the signal should go through the π-network or the through line, two PIN diodes (GC4944s[2]), were used to provide switching depending on the DC bias that represented the bits. As mentioned previously, the "0" bit denoted that the DC bias would be -2.5V, and the "1" bit denoted the DC bias to be 2.5V, causing the PIN diodes to either be forward or reverse biased. If the "0" bit is inputted, the signal would be attenuated by 9.5 dB, the "1" bit represents the signal being fed into the through line for minimal attenuation.

To reduce the number of components, transmission lines were used in place of the inductors in the ideal schematic shown in Figure 12. Figure 13 displays the schematic implemented in ADS. Additional `MLIN` lines were added after the DC bias and `MLOC` transmission line on the through line to help with final circuit tuning.

Figure 14 and Figure 15 display the simulation results for the negative bias and positive bias outputs for the circuit, respectively. It can be seen that the attenuation for the negative bias has an attenuation of roughly 10.5 dB and the positive bias has an attenuation of roughly 0.5 dB. Another simulation output to notice is the polar plot showing the attenuated and non-attenuated signal.

[2]Datasheet: https://ww1.microchip.com/downloads/aemDocuments/documents/RFDS/ProductDocuments/DataSheets/GC4900.pdf

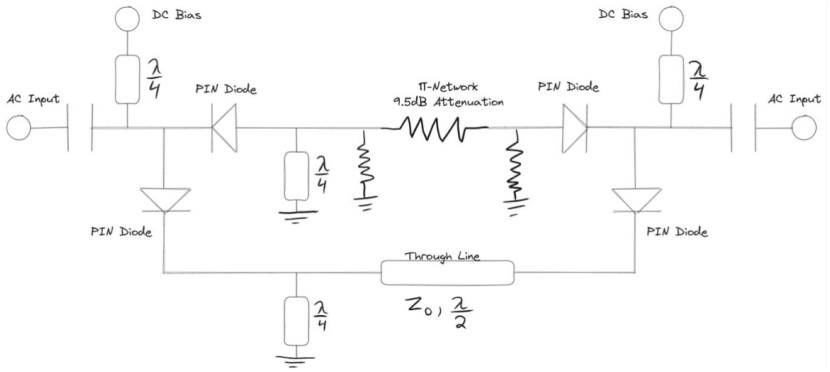

Figure 12: Phase Invariant Attenuator Ideal Schematic

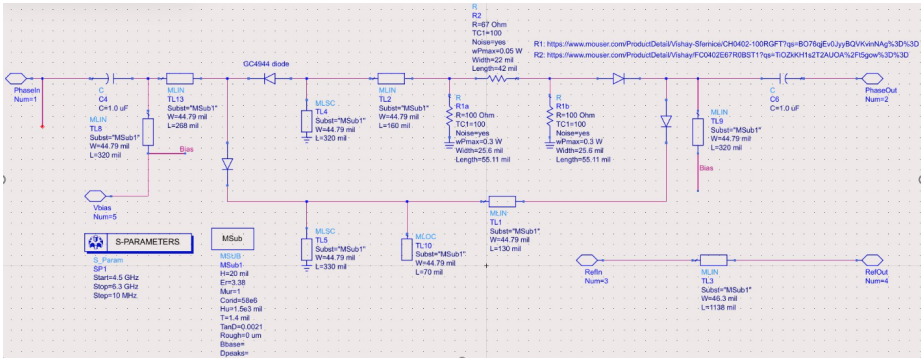

Figure 13: Phase Invariant Attenuator Schematic

While many of these graphs may not seem to hit the requirements (such as the VSWR), in the final implementation, by utilizing `MLOC` lines and more tuning, the final specs were met in the overall design.

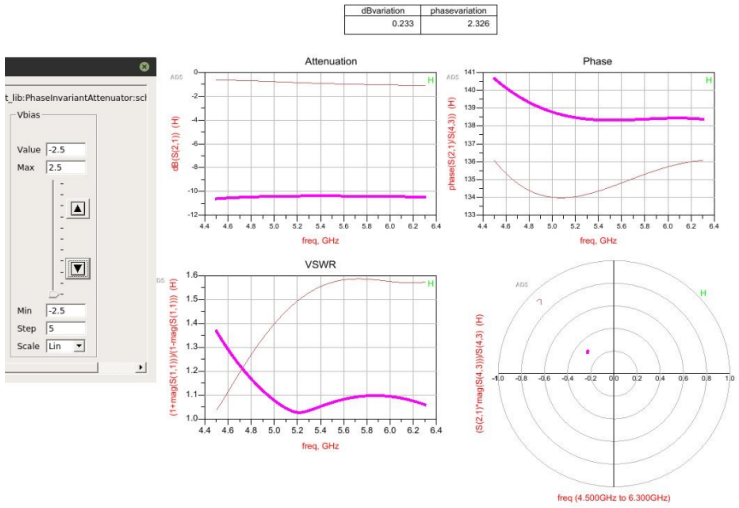

Figure 14: Phase Invariant Attenuator Simulation Negative Bias

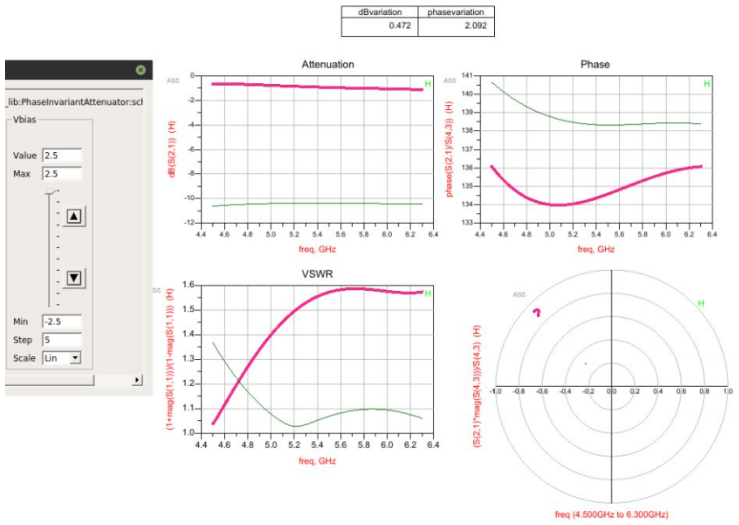

Figure 15: Phase Invariant Attenuator Simulation Positive Bias

3.4 In-Phase Combiner

Finally, the in-phase combiner was achieved with the Wilkinson Power Combiner that was used to combine the power of the two input ports for the benefits mentioned previously. For this stage, there was a lot more focus on the geometry of the design to ensure that with all the transmission lines being used the

design would be physically possible.

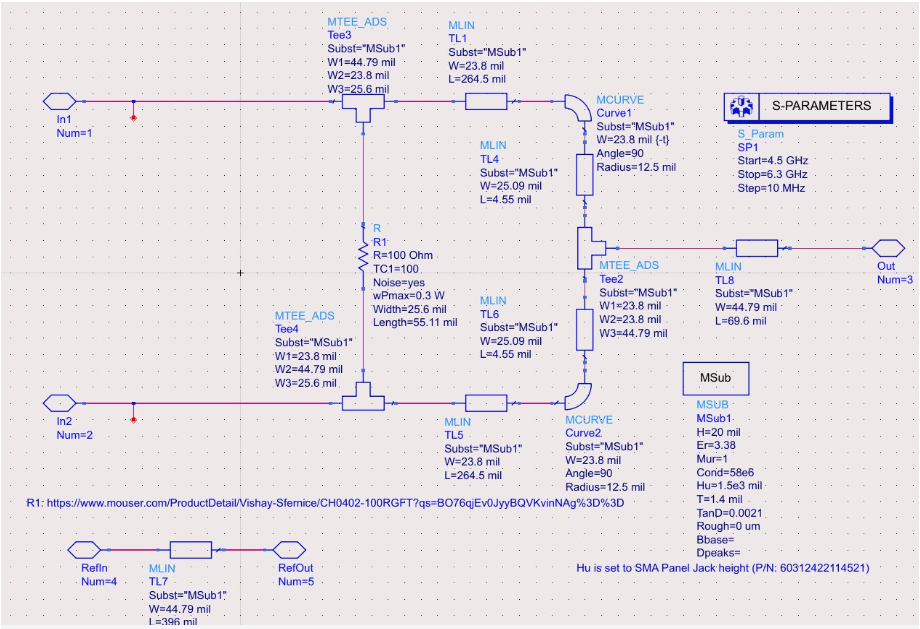

Figure 16: Wilkinson Power Combiner Schematic

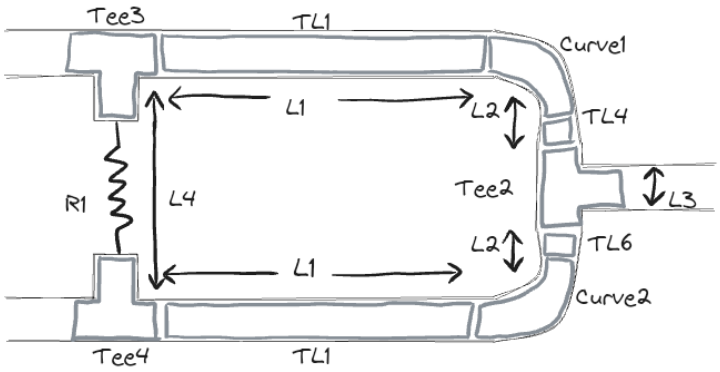

Figure 17: Wilkinson Power Combiner Schematic

First, the length of the 100Ω resistor was determined. By keeping track of the lengths and widths for the various `MLIN`, `MCURVE`, and `MTEE`, these two lengths were matched to be equal. In addition, the top half and bottom half of the Wilkinson power were symmetric. Figure 17 shows a visualization of the values for the different lengths. The total length from the 2 `MCURVE` & `MLIN`, L_2,

and MTEE, L_3, on the right side is equal to the resistor length which is L_4.

The simulation results can be seen in Figure 18. It is shown that the peak output power is 3 dB as expected. It can also be seen that even in isolation, the output phase, phase variation, and VSWR all seem to hit spec for this block. It should also be noted that the resonant peak is a little past 5.4 GHz. This was a result of the tuning process, and the design originally started with resonance at 5.4 GHz.

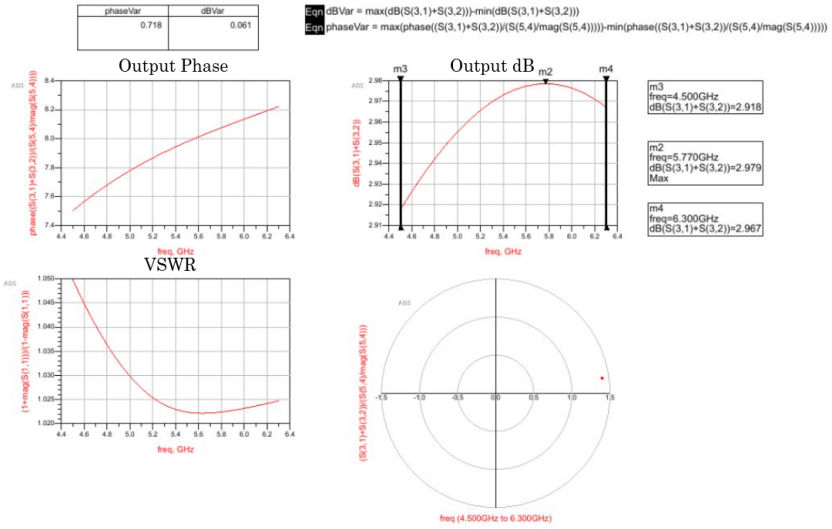

Figure 18: Wilkinson Power Combiner Schematic

4 Results

Figure 19 displays the full schematic done on ADS. The data display tool within ADS was extensively used to make initial judgments of the circuit, before exporting the S-parameter data to CSV files for later post-processing in MatLab. S_{11} and S_{21} were collected. S_{43} represents a transmission line of equal electrical length to the entire vector modulator. It was collected to cancel a linear phase response due to the electrical length of the circuit.

All 16 states for these parameters were exported, and a python script was written with the help of ChatGPT to convert the ADS magnitude/phase format to separate cells of real and imaginary. These were then fed into MatLab to analyze the data and create intuitive graphics. This code is visible in the appendix.

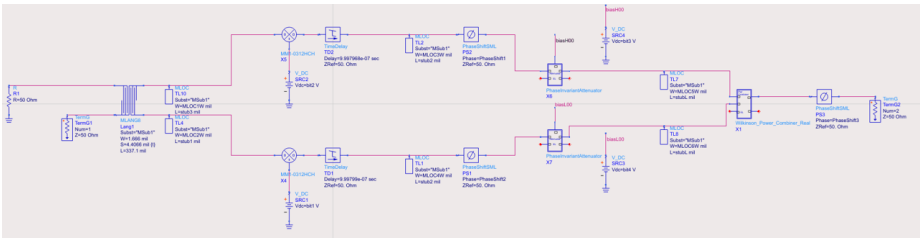

Figure 19: Full Schematic

4.1 The Constellation

To show the constellation (Figure 20), the S_{21} parameters were manipulated to remove the linear phase response and then plotted in the complex plane. In the graph, the ideal points are marked as X, squiggles are the frequency response, and the stars are labeled with their respective states.

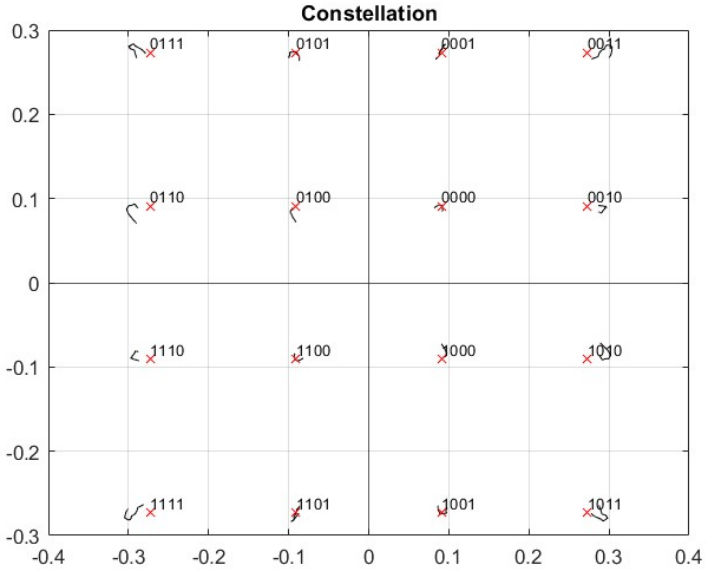

Figure 20: The Constellation

To remove electrical length out of the data, S_{21} was divided by unit magnitude of S_{43} to cancel out the phase:

$$\text{star} = \frac{S_{21}}{S_{43}/|S_{43}|} \tag{1}$$

Because attenuation was considered in the simulation, the magnitude must be removed to obtain a pure phase cancellation.

The ideal points are a matrix of complex points with horizontal and vertical distances normalized to one. As mentioned previously, to have the points be equidistant from each other, the horizontal and vertical spacing is two third for a 16-QAM constellation. The reference point 0000 was chosen, so the entire ideal constellation was scaled to meet that state's magnitude at the center frequency.

It is worth noting that, depending on which state was chosen as a reference, the dB and phase variation became significantly different and pushed more stars out of the design criteria. Since state 0000 had the best specs overall, it was chosen as the reference point.

4.2 Worst-Case dB and Phase Error

To intuitively show what points met the requirements versus what points did not, a phase-dB plane (Figure 21) was created, with each point's worst-case deviation plotted. For both the dB and the phase worst-case scenarios, each state's S_{21} was swept with respect to the state's ideal point. This is shown in equations (2) and (3). The ± 0.7 dB and ± 3.5 degree phase variation specifications, allows us to place a box at these coordinates and easily visualize which stars met the design criteria.

$$\text{Max dB Deviation} = \max \left| \, 20 \log |\text{ideal}| - 20 \log |S_{21}| \, \right| \tag{2}$$

$$\text{Max Phase Deviation} = \max \left| \, \text{phase}(\text{ideal}) - \text{phase}(S_{21}) \, \right| \tag{3}$$

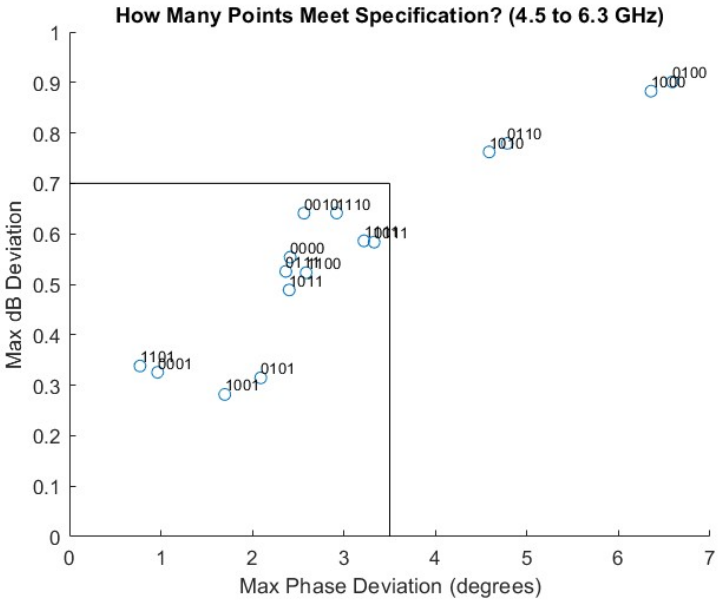

Figure 21: Each star's worst case phase and dB variation.

There are several things to notice when looking at the plot. First and foremost, four points fail to meet specification. More specifically, the 1010, 0110, 1000, and 0100 states. These fail both the dB and phase-variation requirements by a significant amount. Something else to notice, is that the points seem to appear in pairs on this chart, and that is a result of the 180° symmetry of the constellation from Figure 20. Looking back and checking these pairs of states, the squiggles do also exhibit this symmetry, with some slight translations that are a result of unmatched paths in the vector modulator.

4.3 VSWR

Each state's VSWR is shown in Figure 22. These were calculated using the formula:

$$\text{VSWR} = \frac{1 + |S_{11}|}{1 - |S_{11}|} \tag{4}$$

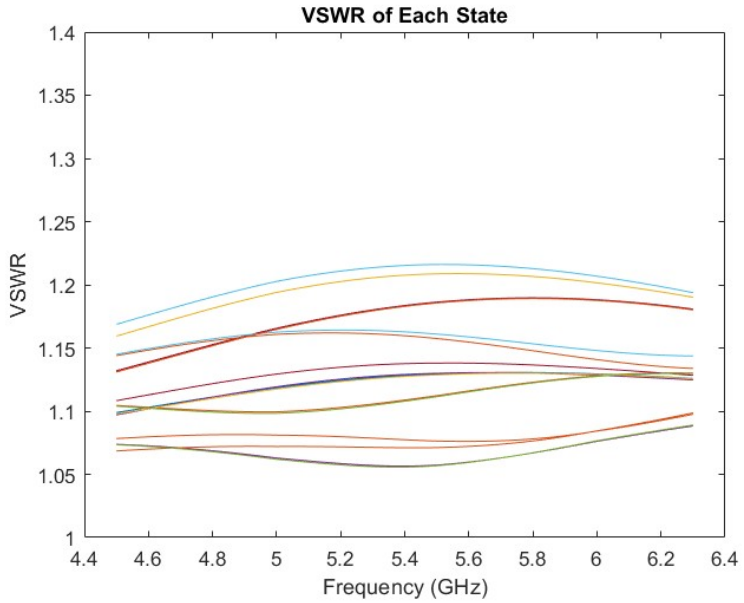

Figure 22: Each star's VSWR over the 1.8 GHz bandwidth.

The specification required a VSWR of less than 1.4, and this was illustrated in the figure by setting the upper bound to 1.4. Evidently, no state crosses this boundary at any point in the bandwidth.

4.4 Insertion Loss

Insertion loss was calculated for only the outermost states, i.e. 0011, 0111, 1011, and 1111. These states must have no more than 8.5 dB of insertion loss,

meaning their respective $-20\log|S_{21}|$ must be less than 8.5 dB. Insertion loss is plotted in Figure 23.

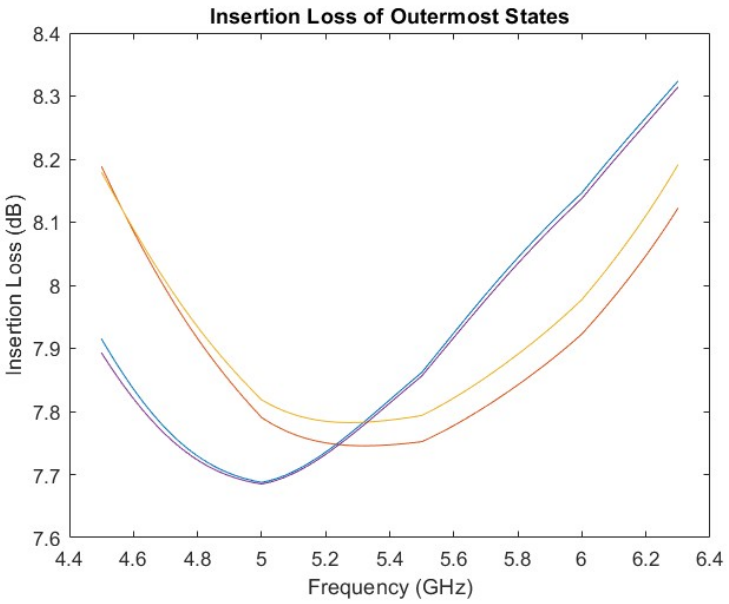

Figure 23: Insertion loss of outermost stars.

4.5 Table of Results

State	Achieved Error	Margin
"0000"	2.4120	1.0880
"0001"	0.9643	2.5357
"0010"	2.5631	0.9369
"0011"	3.3292	0.1708
"0100"	6.5879	-3.0879
"0101"	2.0904	1.4096
"0110"	4.7842	-1.2842
"0111"	2.3647	1.1353
"1000"	6.3550	-2.8550
"1001"	1.6976	1.8024
"1010"	4.5859	-1.0859
"1011"	2.4007	1.0993
"1100"	2.5865	0.9135
"1101"	0.7714	2.7286
"1110"	2.9188	0.5812
"1111"	3.2179	0.2821

Table 1: Table of phase variation for each state. Negative numbers mean that the state failed the $\pm 3.5°$ specification.

State	Achieved Error	Margin
"0000"	0.5536	0.1464
"0001"	0.3257	0.3743
"0010"	0.6409	0.0591
"0011"	0.5836	0.1164
"0100"	0.9012	-0.2012
"0101"	0.3146	0.3854
"0110"	0.7793	-0.0793
"0111"	0.5256	0.1744
"1000"	0.8827	-0.1827
"1001"	0.2817	0.4183
"1010"	0.7623	-0.0623
"1011"	0.4888	0.2112
"1100"	0.5231	0.1769
"1101"	0.3381	0.3619
"1110"	0.6413	0.0587
"1111"	0.5861	0.1139

Table 2: Table of dB variation for each state. Negative numbers mean that the state failed the ± 0.7 dB specification.

	Required Spec	Achieved Spec	Margin
Max VSWR	1.4	1.22	0.18
Max Insertion Loss	8.5 dB	8.32 dB	0.18

Table 3: Table Displaying the Maximum VSWR and Insertion Loss.

5 Other Considerations

There were other considerations of the circuit that we had to take into account if it was built in real life. The size, weight, cost, and power consumption were all important factors and design decisions.

In terms of size, the width and length of the MLANG8 in the quadrature divider was 1.666 mils and 337.1 mils, respectively. This was added to the size of the mixer chip, which was 1.81 mils wide and 2.13 mils long, as seen in the figure below. The total size of the Phase Invariant Attenuator 179.16 mils wide and 2608 mils long. For the Wilkinson Power Divider, the length was 607.7 mils and the width was 47.6 mils. The total size of the resistors was 162 mils by 118 mils. The weight of the circuit including the Rogers 4003, resistors, and Mixer chip will be around 3.3g.

The cost of the circuit was carefully considered. All of the components were relatively cheap, except for the Marki Mixer component, costing $64.50. This was the most expensive part of the circuit.

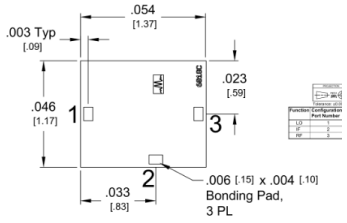

Figure 24: Size of MM1-0312HCH

6 Future Improvements

The biggest challenge when meeting all of the specifications was the insertion loss that each block experienced. The highest source of insertion loss was the mixer, which had experienced an insertion loss of around 7.5 dB. Since the mixer was a real part, it was chosen for its lowest insertion losses out of the other mixers produced by Marki. Finding a better mixer from another manufacturer would allow more room for fixing other parameters without having to worry about the insertion loss trade off.

The next improvement would be to go back and correct the schematic for how the DC bias is handled for the phase invariant attenuator circuit. While designing the schematic, it wasn't taken into account how much current each of the diodes can handle, only whether it was forward biased or reversed biased, so it is likely the diodes wouldn't work in a real design. In addition, the way the DC bias was fed to the diodes can be done better where the DC and RF signals are kept isolated from each other. Finally, there should be additional capacitors on the shunt resistors in the π-network so that the DC side is kept isolated from the AC side.

Another improvement is to make sure the phase shifters on the top and bottom side of the circuit are kept exactly the same since those are ideal components meant for visualizing the data better, rather than to help tune the circuit. They're very close right now and it could be misinterpreted that the phase shifters were tuned to make the circuit perform better than it actually should.

In addition, for multiple parts of the design, there were ideal capacitors and inductors used instead of using real capacitor and inductor value which have package parasitic values. Adding these new parts would mean the circuit would need to be retuned, but would make the overall design more realistic. Another step to take would be to consider the geometry of the transmission for all the stages similar to how it was done for the Wilkinson Power Combiner.

Finally, the last improvement is to view the points that missed spec from different perspective when trying to determine out where to improve the circuit so all the points hit spec. Since all the points hit spec at 5.4GHz, but fall out of spec as the frequency changes, one way to view the missing spec is that the bandwidth of the circuit was too small. By having this perspective, it's possible to go back to the different blocks and increase the bandwidth, such as having an over coupled Lange coupler to increase the overall bandwidth of the circuit.

7 Conclusion

Overall the design didn't meet all of the specifications. Specifically the 4 states (1010, 0110, 1000, and 0100) didn't meet the dB and phase variation. There are also several steps needed to make this design much more realistic and closer to a real life that can be implemented in real life. Despite all this, this was a rather successful first attempt at making the 16-QAM Vector Modulator. 12 points did meet these specifications set and all the points did meet the maximum insertion loss and VSWR values.

This project highlighted how much consideration must be given when working with cascaded stages since individual may seem to be working (or even not working), but when the design is connected together the system behaves differently. In addition, it also highlights a lot of the design choices that has to be considered for a real design to be implemented that is often ignore when running a much simpler simulation. Some of these choices include the physical constraint to make the design fit, or determining a better way to do data analysis instead

of sticking to only to ADS.

8 Appendix

8.1 Datasheets:

- Mixer: https://markimicrowave.com/products/surface-mount/mixers/mm1-0312hsm-2/datasheet/
- PIN Diode: https://ww1.microchip.com/downloads/aemDocuments/ documents/RFDS/ProductDocuments/DataSheets/GC4900.pdf
- Ra: https://www.vishay.com/doc?53014
- Rb: https://www.vishay.com/doc?60093
- R1: https://www.vishay.com/doc?53014
- R1: https://www.vishay.com/doc?53014
- 50 ohm Termination Resistor: https://www.vishay.com/doc?53014
- Rogers 4003: https://www.rogerscorp.com/advanced-electronics-solutions/ro4000-series-laminates/ro4003c-laminates

8.2 Python Code:

```
import pandas as pd
import os
import math
import re

# Folder containing the files
input_folder = r"inputFilePath"
output_folder = r"outputFilePath"

# Loop through all CSV files in the folder
for filename in os.listdir(input_folder):
    if filename.endswith(".csv"):  # Process only CSV
        files
        file_path = os.path.join(input_folder,
            filename)
        df = pd.read_csv(file_path, header=None)  #
            Load the CSV file

        # Process column A: Remove letters from values
            starting from row 2
        cleaned_col_a = []
        for index, value in enumerate(df[0]):
            if index >= 1:  # Start from row 2
                if isinstance(value, str):  # If the
                    value is a string
                    cleaned_value = re.sub(r'[a-z,A-Z]
                        ', '', value)  # Remove all
                        letters
                    cleaned_col_a.append(cleaned_value
                        .strip())  # Strip leading/
                        trailing spaces
                else:
                    cleaned_col_a.append(value)
            else:
                cleaned_col_a.append(value)  # Keep
                    header or row 1 as is

        # Replace column A with cleaned values
        df[0] = cleaned_col_a
        # Process magnitude/phase columns (B, C and D,
            E)
        for col in range(1,5):  # Columns with
            magnitude (B and D)
            #print(col)
```

```
        real_parts = []
        imag_parts = []

        for index, row in df.iterrows():
            if index >= 0:  # Start from row 2
                try:
                    # Split the value into
                        magnitude and phase
                    magnitude, phase = row[col].
                        split('␣/␣')
                    #print(float(phase), phase,
                        "|", float(magnitude),
                        magnitude)
                    # Convert phase to radians and
                         calculate real/imaginary
                        parts
                    phase_rad = float(phase)*math.
                        pi/180  # Convert phase to
                        radians
                    real_parts.append(float(
                        magnitude) * math.cos(
                        phase_rad))
                    imag_parts.append(float(
                        magnitude) * math.sin(
                        phase_rad))
                except Exception as e:
                    # In case of any error (e.g.,
                        bad data format), append
                        None
                    #print(f"Error processing row
                        {index} in column {col}: {e
                        }")
                    real_parts.append(None)
                    imag_parts.append(None)

        #print(real_parts)
        # Add real and imaginary parts as new
            columns
        real_col = f"Real␣({col+4})"
        imag_col = f"Imaginary␣({col+4})"
        #print(real_parts.append(0))
        df[real_col] = real_parts
        df[imag_col] = imag_parts
        #print(df[real_col])

    #for i in range(1,7):
```

```
            #     df[i] = df[i+4]
            #     df[i+4] = None
            # Save the updated file as CSV
            output_path = os.path.join(output_folder,
                filename)
            df.to_csv(output_path, index=False)

print("Processing␣complete!")
```

8.3 MatLab Code:

```
%% star plots

Re = -1:2/3:1;
Im = -1:2/3:1;
[Re, Im] = meshgrid(Re, Im);
idealStars = (Re-1j*Im);
idealStars = (3/sqrt(2))*abs(S{1}(91,2))*reshape(
   idealStars,1,[]);
%uses 0000 as a reference
clear Re Im
freq = 4.5:0.01:6.3;
states = ["0000" "0001" "0010" "0011" "0100" "0101"
   "0110" "0111" "1000" "1001" "1010" "1011" "1100"
   "1101" "1110" "1111"];
statesInOrder = [states(8) states(7) states(15) states
   (16) states(6) states(5) states(13) states(14)
   states(2) states(1) states(9) states(10) states(4)
   states(3) states(11) states(12)];

figure
for star = 1:L
    plot(S{star}(:,2)./(S{star}(:,4)/abs(S{star}(:,4))
       ),"-k");
    hold on
end
plot(idealStars, "xr")
text(real(idealStars), imag(idealStars), statesInOrder
   , 'Vert','bottom', 'Horiz','left', 'FontSize',7)
title("Constellation")
xline(0)
yline(0)
grid()
hold off

% phase and dB variation

smallrange = 30:151; %4.79 to 6 GHz
range = 1:181;
dBVar = zeros(1,L);
phaseVar = zeros(1,L);
inOrder = [idealStars(10) idealStars(9) idealStars(14)
    idealStars(13) idealStars(6) idealStars(5)
   idealStars(2) idealStars(1) idealStars(11)
   idealStars(12) idealStars(15) idealStars(16)
```

```
    idealStars(7) idealStars(8) idealStars(3)
    idealStars(4)];

for star = 1:L
    phaseVar(star) = max(abs(rad2deg(angle(S{star}(
        range,2)./S{star}(range,4)))-rad2deg(angle(
        inOrder(star)))));
    dBVar(star) = max(abs(20*log10(abs(S{star}(range
        ,2)))-20*log10(abs(inOrder(star)))));
end

figure
scatter(phaseVar, dBVar)
title("How Many Points Meet Specification? (4.5 to 6.3
    GHz)")
line([3.5,3.5], [0,0.7],"color", 'k')
line([0,3.5],[0.7,0.7],"color", 'k');
xlabel("Max Phase Deviation (degrees)")
ylabel("Max dB Deviation")
text(phaseVar, dBVar, states, 'Vert','bottom', 'Horiz'
    ,'left', 'FontSize',7)

%VSWR
figure
for star = 1:L
    plot(freq, (1+abs(S{star}(:,1)))/(1-abs(S{star
        }(:,1))));
    hold on
end
title("VSWR of Each State")
xlabel("Frequency (GHz)")
ylabel("VSWR")
ylim([1 1.4])

%Insertion Loss
figure
for star = 4:4:16
    plot(freq, -20*log10(abs(S{star}(:,2))));
    hold on
end
title("Insertion Loss of Outermost States")
xlabel("Frequency (GHz)")
ylabel("Insertion Loss (dB)")
```