A 12-bit, 500 MHz, current steering DAC for use in an FMCW radar system

by

Bo E. Lindstrom

B.S., Kansas State University, 2019

A THESIS

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Mike Wiegers Department of Electrical and Computer Engineering
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2020

Approved by:

Major Professor
Don M. Gruenbacher

Copyright

Abstract

Digital-to-analog converters are an essential part of modern electronic systems which demand low-power consumption, high-speed performance, and exceptional linearity. They have a seemingly infinite number of applications, and as commercial integrated circuit processes continue to delve further into deep sub-micron territory, their utility, performance, and efficiency will only improve.

This thesis focuses on the design and implementation of a 12-bit, 500 MHz, current-steering DAC in a 45nm SOI CMOS process. The DAC was designed using a fully-segmented, or 'thermometer-coded' topology, and will be fabricated in the aforesaid process. Ultimately, the DAC is destined for integration in a K-band stepped-frequency FMCW radar system, where it will act as an intermediary between a direct-digital-synthesis submodule and a phase locked loop submodule. – The DAC will convert the DDS produced discrete-time sinewave output, which has a low over-sampling ratio, to an analog continuous time waveform suitable for bandpass filtering. Bandpass filtering then provides time-smoothing to improve phase noise and spurious levels from the PLL.

To fully realize a workable DAC, several other sub-systems exist which contribute to its overall function. The design and implementation of these sub-systems – digital thermometer decoders, D-Flip-Flops, and individual current cells – will be described in detail. Trade-offs and challenges encountered throughout the 'creation-story' of this system are also discussed.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Notice: This manuscript has been authored using funds under the Honeywell Federal Manufacturing & Technologies under Contract No. DE-NA-0002839 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

Chapter 1 - Introduction

1.1 Motivation

For the past few years, a research team at Kansas State University has been developing a K-band stepped frequency radar for agricultural applications. This radar is currently implemented through a number of submodules consisting of COTS, ICs, and discrete components on an FR4 laminate printed circuit board. It has evolved through several revisions since its inception, and has been put through numerous laboratory trials, and an extensive set of field tests in order to verify; and continually improve upon its performance.

Within the last year and a half, a shift in focus has occurred – in lieu of continuing to develop further revisions of the K-State Ag-Radar as it is currently implemented (discretely on a printed circuit board), efforts have been made to develop an integrated circuit which matches or exceeds the functionality of the discrete radar.

Implementation of the K-State Ag-Radar on an IC chip offers several advantages. First and foremost, it miniaturizes the whole system. This allows for a physically lighter, lower-power design – one which would open new doors for applications previously unattainable with the discrete radar. Second, it affords Kansas State students the opportunity to gain experience in mixed signal IC design flow. This not only creates research and publication opportunities for faculty and students, but also develops a knowledge base and student skill-set that the sponsors of the K-State Ag-Radar may find desirable.

One of the essential components of the miniaturized radar is a high-speed digital-to-analog converter to interface between a direct digital synthesis submodule and another submodule containing a phase-locked-loop. The main body of work highlighted in this document pertains to the design and verification of a DAC for this specific application.

In an ideal world, this document would have focused not only on the digital-to-analog converter, but also the direct digital synthesis submodule as well. The original intent of the research team was to develop these two systems in tandem, and complete and verify a full mixed-signal design through simulation. Of course, this mixed signal design would then be fabricated and put through another rigorous round of testing and verification. As it stands now, however, the digital libraries needed to complete a true mixed signal design flow are unavailable to the research team - only work on the digital-to-analog converter has been completed.

The K-State Ag-Radar IC is currently being developed in a 45nm SOI process. In early May of 2020, the DAC and several other submodules of the radar were sent out for fabrication – this was funded by K-State's sponsors, the Kansas City National Security Campus (Honeywell). Honeywell has sponsored a consortium of land-grant universities in the Midwest for several years, and facilitates research and collaboration through these schools, namely University of Arkansas, University of Kansas, Kansas State University, Michigan State University, University of Nebraska-Lincoln, and University of Oklahoma. The research teams funded by this consortium specialize in high-frequency (RF) electronics design, and the group is appropriately named the 'Radar Consortium'.

The research team expects to receive the DAC IC in the fall of 2020, upon which it will be tested extensively. There are sure to be interesting results – this particular 45nm process is unfamiliar to both students and faculty at K-State. That said, the design presented in the following chapters has been extensively simulated and is expected to perform well, with clocking speeds of 500 MHz or greater. Details of the design and verification are the focus of this thesis.

1.2 Prior Work

Current steering digital-to-analog converters have been in use since the 1970's [1], and became commonplace in the 1980's with descriptions of 8 and 12 bit partially segmented (a hybrid topology combining binary weighted current cells and unary weighted current cells) variations of this topology appearing as early as 1983 [2], [3]. A year later, DAC's of 8, 10, and 12 bits were reported which were fabricated in a 3um bipolar process and could operate at 40 MHz, with a supply of 5 V [4].

As IC processes continued to shrink, performance and power consumption improved drastically. By the end of the 1990's, operating frequencies of 500 MHz were commonplace, supply voltages had dropped to 3.3 V, and DACs were being fabricated in processes that sized below 0.4um [5]. It was now possible for designers to fabricate 12-bit partially segmented DACs that consumed a die area less than 3.5 mm^2 with a total power consumption less than 0.5 W [6]. As technology improved, demand increased accordingly, and DAC designers continued to push the state-of-the-art. The turn of the century saw engineers armed with improved IC processes such that they could continue to build on the successes and knowledge of their predecessors.

In the 2000's, supply voltages dropped under 2 V – this allowed for continued improvements in power consumption, and with smaller and smaller processes, speed and total chip-area improved significantly [7]. By the end of the decade, 12-bit DACs were being designed in deep sub-micron processes, operating near 3 GS/s, and consuming less than 200 mW [8], [9]. This was an incredible leap forward in performance considering where technology stood at the end of the 1990's.

Total power consumption continued to improve in the 2010's, with figures in the realm of 35 mW reported for 12-bit designs [10]. Total area consumption below 2500 um^2 for 12-bit designs

were also reported [11] – again, a significant improvement, made possible by deep sub-micron processes. Reports of successful 12-bit DAC designs in a 40nm process – a similar size to the node used by the K-State team – were published in 2016. These designs were powered on a DC supply voltage of 0.9, and consumed a scant 1.5 mW of power [12]. This DAC, however, was implemented using capacitors and did not follow a current steering topology. It consisted of an array of unit capacitors, and an additional section used for calibration – a 9-bit binary weighted array [12]. The use of capacitors is a relatively unique choice for modern DAC design. The device in question was interfaced with an ADC to provide calibrations contributing to superior performance and linearity, and that specific application may have informed the designer's choice of topology.

More recently, in 2019, another 12-bit design was reported in a 40nm process which consumed a mere 50.8 uW of power, and consumed only 270 um^2 of area [13]. The authors reported a maximum clock frequency of 900 MHz, but noted that the optimal clock frequency in terms of a power-resolution tradeoff came much lower, at 450 MHz [13]. This DAC uses dyadic digital pulse modulation (DDPM) to achieve data conversion between the digital and analog domains, and the authors describe designing the DAC using a fully automated digital design flow as opposed to the typical 'by-hand' approach taken by analog designers working in the field of data conversion [13].

Today, speeds of well over 1 GS/s are the norm [14], and 10 GS/s are not out of the ordinary for 12-bit designs [15], and partially-segmented current steering DACs of 12-bit resolution are now being designed in processes as small as 14 nm [16].

To the best of the author's knowledge, the design that will be presented in this thesis is one of the only, if not the only, fully thermometer coded 12-bit DAC to see fabrication, and certainly the only in a 45nm process.

1.3 Main Contributions

The design and verification, through simulation, of this DAC made two key contributions to the field of data conversion.

1. A fully thermometer-coded 12-bit DAC in a 45nm SOI process.
2. A list of 'lessons learned' – most significantly with regard to the simulation of digital-to-analog converters, but also those which pertain to sound design and layout practices for large, hierarchical designs.

1.4 Thesis Organization

An overview of the Agricultural radar system this DAC will interface into can be found in Chapter 2. That same chapter also includes some remarks on the digital designs which will find their way into future revisions of the IC, upon receiving the requisite libraries. Chapter 3 gives a thorough description of the different design approaches one may take to implement a digital-to-analog converter. It also gives a brief discussion to the advantages and disadvantages of each topology. Chapter 4 contains insights into the specific design details of the DAC and surrounding sub-systems. Chapter 5 includes simulation results of the circuit in question, and Chapter 6 consists of conclusions, lessons learned, future plans, and closing remarks.

Chapter 2 - The Kansas State Agricultural Radar

2.1 System Background

The K-State Agricultural Radar, in which the DAC will be used, is a stepped frequency FMCW system operating in K-band, at 23.5-26.5 GHz. This frequency range was chosen due to the fine resolution it provides for close range observation of plants [17].

Ultimately, the Ag-Radar could be a valuable tool for farmers and academics embroiled in crop-research, as it can provide data on crop-height, density, and moisture content. Currently, the system is implemented on an FR-4 laminate PCB, and field tests have been conducted with the radar 'looking' out the window of a motor-vehicle as it makes passes through a row of crops. Figure 1, below shows the current revision of the radar system designed by Garrett Peterson within the research team at K-State [17].

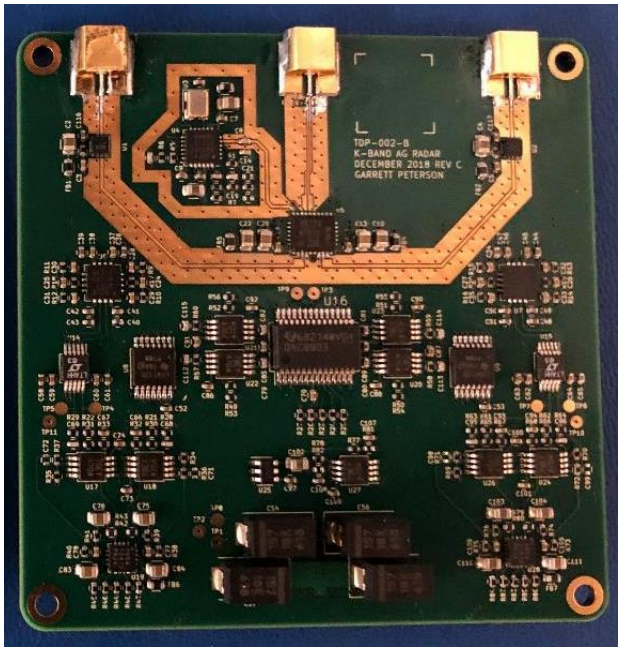


Figure 1 *– Rev. C of the K-State Agricultural Radar [Garrett Peterson, Summer 2019, Manhattan, KS]*

Figure 2 shows the board mounted in a protective case, along with 3-D printed horn antennas.

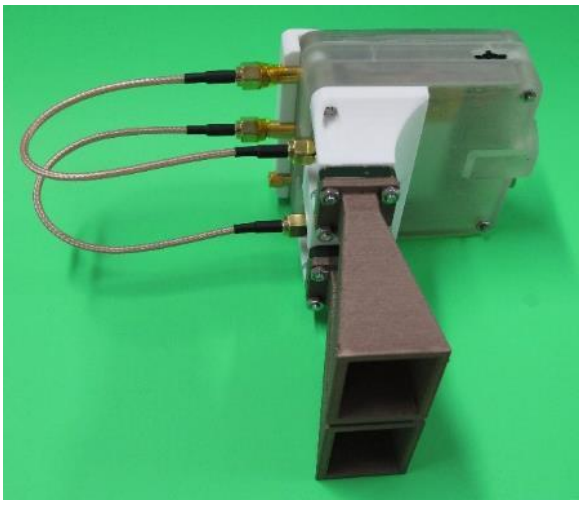


Figure 2 *– The full K-State Ag-Radar as it is would appear during a field trial [Garrett Peterson, Summer 2019, Manhattan, KS]*

Figure 3 shows the current method of testing the radar – driving through rows of crops with the radar mounted on a pole 'looking' down on the crops.

Figure 3 *– The K-State Ag-Radar mounted to stand and 'looking' out the window of a truck into a row of crops [Garrett Peterson, Summer 2019, Manhattan, KS]*

Miniaturization would allow the Ag-Radar to be implemented in a much more convenient manner, decreasing size, weight, and power (SWaP). This could facilitate mounting to a small drone, for example, which would allow for much more extensive data collection in a timelier manner. It is for this reason that work commenced on implementing this system at the integrated circuit level. Figure 4 below contains a block diagram showing the basic architecture of the proposed integrated, miniaturized radar system.

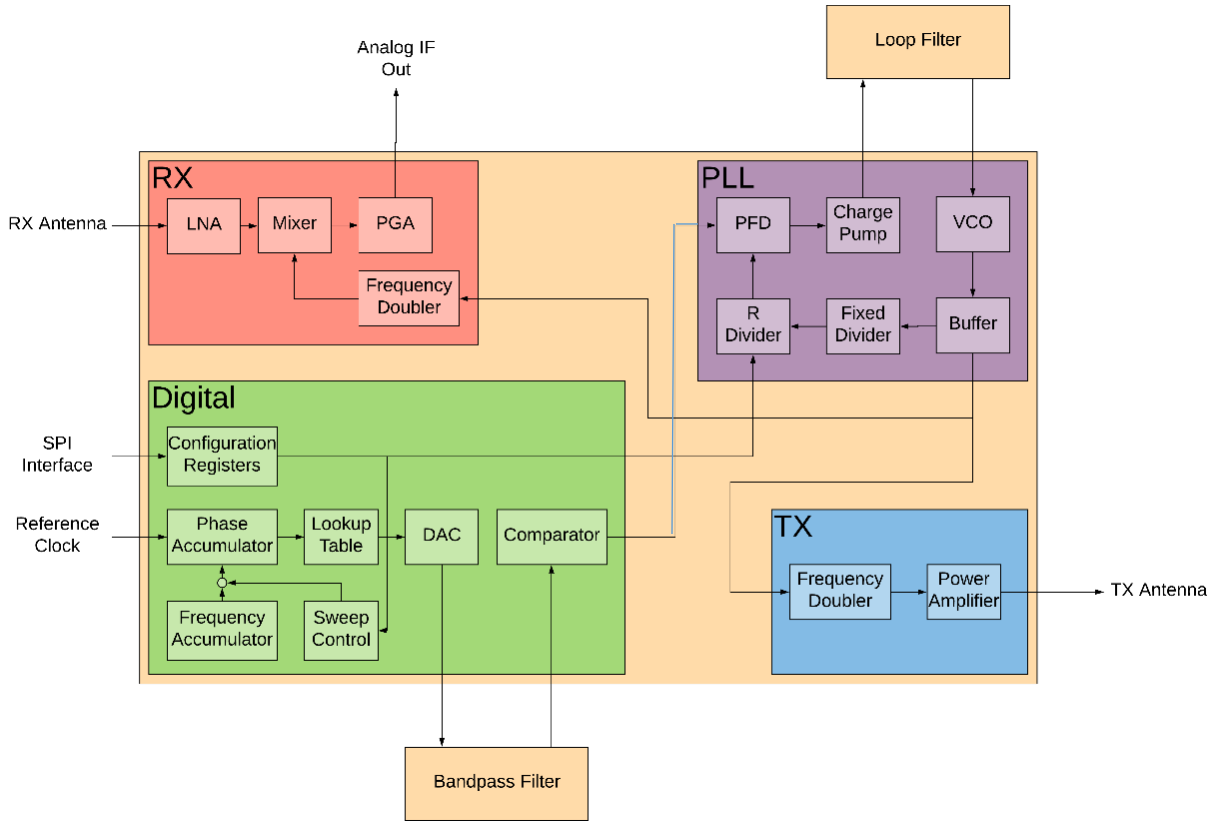


Figure 4 *– A block diagram showing the submodules and approximate floorplan of the K-State Ag-Radar*

2.2 Digital Block

Starting outside the lower left-hand block, an SPI interface shifts in data from off-chip to configure the sweep parameters of the RADAR's frequency-modulated CW signal. The purpose

of the digital block is to generate a low-jitter swept reference frequency squarewave for use in the integer-N PLL, which is contained in the upper right-hand block of the figure. The PLL upconverts the swept frequency to the 11.8 to 13.3 GHz range and supplies it to the TX and RX blocks to form a complete FMCW radar operating at 23.5 to 26.5 GHz.

The swept frequency reference signal is generated by a direct digital synthesis (DDS) architecture consisting of a phase accumulator, and associated frequency accumulator and control blocks with the digital submodule. The phase-ramps from this DDS are then converted to a discrete-time sinewave in the lookup table and delivered to the DAC.

The digital-to-analog converter creates a ‘stair-case’ like representation of the discrete-time sinewave. This coarse-shape sinewave is then fed to an off-chip LCR bandpass filter to create a more continuous, smooth, and spectrally pure representation of the sinewave. Finally, the signal, now recognizable as a sinewave, is sent to a comparator to produce a low-jitter square wave for use in the PLL.

2.3 Jitter Analysis and Need for Filtering

The filtering of the DAC output off-chip, and the use of a comparator to square up the resulting sinewave serves to reduce jitter, and suppress spurious tones in the waveform which drives the PLL [18]. This process ensures that the signal arriving at the PLL is one of acceptable purity.

Jitter in the DDS output is of chief concern when quantifying the performance of such a system, and has been the focus of much academic inquiry. It is generally accepted that seven sources of jitter exist in DDS systems [19]. These are input clock jitter, finite bit resolution of the phase accumulator, finite bit resolution of the look-up table, finite bit resolution of the DAC, the accuracy of the DAC, characteristics of the analog filter, and the accuracy of the comparator.

Input clock jitter is expressed as a function of the time differences between edges of the real clock and edges of an ideal clock with no jitter. It is very difficult to completely rectify jitter on the input clock, however the jitter this non-ideality contributes to the output waveform is of a lesser magnitude than the jitter on the input clock. Therefore, it can be concluded that the SNR of the output of a DDS can be superior to that of the SNR of the clock.

Jitter contributions from the accumulator occur due to inconsistencies in the value the register 'resets' to each time the resultant of the control word and present value of the register exceed the maximum value the accumulator can resolve. Ideally it would reset to zero, but this does not always occur. The accumulator may 'roll-over' to a different value after each pass through the look-up table. These frequency errors, which can be interpreted as jitter, are known to increase with time.

The finite resolution of the look-up table contributes to phase errors. In this case, the DAC converts $\sin(\Phi)$ to $\sin(\Phi - \Phi_\varepsilon)$, where Φ_ε is a phase error.

The finite resolution of the DAC causes a 'parasitic' signal with a maximum amplitude of ½ LSB and undetermined frequency to be present at the output. From a system level, it has been shown that the resolution of the input argument to the look-up table should be 1 bit greater than the resolution of the DAC to ameliorate the magnitude of these parasitics if they prove to contribute to unacceptable magnitudes of jitter.

The precision of the DAC can contribute to jitter as glitches occur during bit transitions. Additionally, nonlinearities in the DAC output can compromise the effective resolution of the device. Compensations for this nonlinearity can be made in the look-up table, however.

Analog filtering can suppress unwanted frequency components. The unwanted component of the spectrum which is nearest to the fundamental is the hardest component to

suppress, but also the most important to suppress in order to assure an acceptably pure signal. The authors of [19] propose a series of equations to aid in filter design which account for the ratio of the magnitude of the desired tone and the spurious elements in the signal.

2.4 Lab Demonstration of Analog Signal Processing for a DDS System

This process was demonstrated in lab tests by the author. A DDS was synthesized on a Cyclone IV E FPGA and then fed to a 12-bit DAC. The DAC output was then filtered and squared up using a Schmitt Trigger comparator. Figure 5 displays a lightly oversampled (2.5x Nyquist) sinewave output from a DAC, as well as a square wave output attained by observing the MSB of the phase-accumulator. Figure 6 a case with twice the oversampling ratio (5x Nyquist). The lower curve shows the jitter from the raw phase-accumulator-derived squarewave and the upper trace shows a low-pass filtered, time-smoothed DAC output. This sine wave clearly shows no observable jitter, and will produced a low-jitter square wave after passing through the comparator in Figure 4 on its way to the PLL block.

Figure 5 *– Unfiltered DAC output*

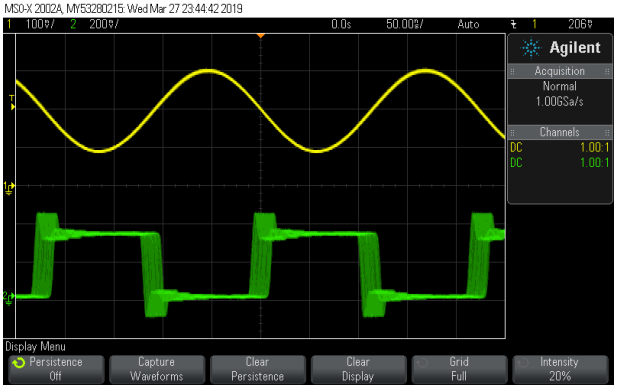


Figure 6 *– Filtered DAC output*

2.5 Further Submodules of the Ag-Radar IC

This square wave signal is delivered to the PLL where it is converted to a much higher frequency (11.8 – 13.3 GHz) through integer-N frequency synthesis. The high frequency output

is then sent to the transmit block where it is brought up to K-band (23.5-26.5 GHz). The transmit block is shown in the lower right-hand corner of Figure 4.

In the upper left-hand corner of Figure 4, the receive block is shown. Following FMCW radar design practice, the received signal is mixed with the transmit signal down to a baseband sinewave whose frequency encodes the target's range. This baseband waveform is then sent off chip to a DSP system for interpretation and analysis.

2.6 Theory and Fundamentals of Direct Digital Synthesis

Direct digital synthesis (DDS) is the process whereby a signal is generated via digital logic and a look-up table or memory block. Its function is easy to understand and is detailed in the figure below.

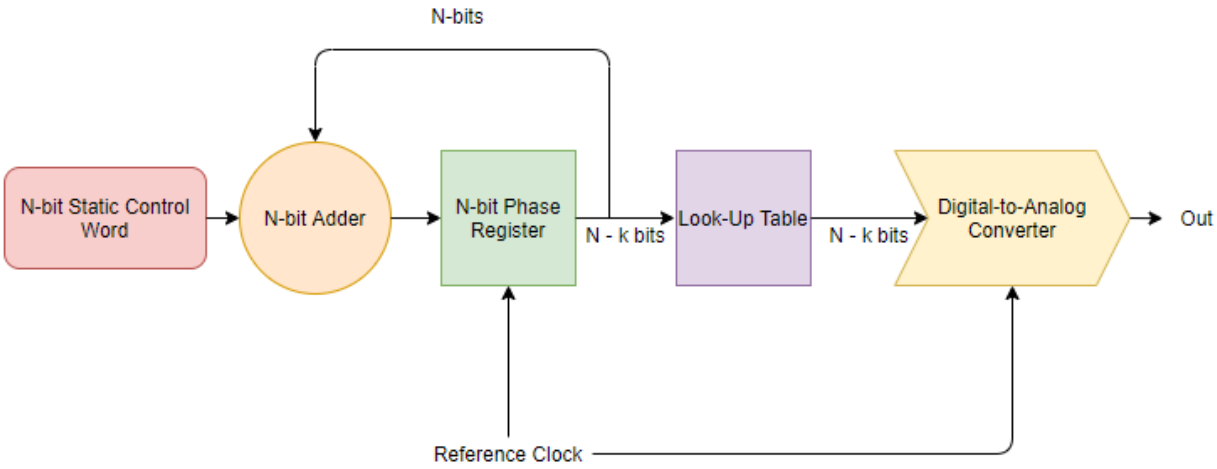


Figure 7 *– A basic block diagram of a DDS system*

A DDS starts with a static control word which is fed into an adder on the first clock cycle. That adder outputs to a register – the 'Phase Register' as it is labeled in the figure above. This register will accumulate every clock cycle as its value will be fed back to the adder. It will then be summed with the static control word. Additionally, a truncated value (truncation is not mandatory but commonly used in high resolution systems) of the Phase Register is sent as an

address to a look-up table or memory block. The resolution of this truncated value has a direct effect on the phase truncation error of the system.

Phase truncation has no effect on the frequency resolution of the system. It is merely used to reduce the size and power consumption of the look-up table, but can introduce spectral impurities into the output of the DDS.

This look-up table generally contains amplitude data for a waveform such a sine, sawtooth, or triangle wave (though any arbitrary waveform could be encoded within). The resolution of the address is generally the same as the resolution of the data contained within the lookup table, although this is not always true. The resolution of the address determines the phase quantization error, and the number of bits in each word contained within the ROM will determine the amplitude quantization error.

The output of the look-up table then feeds a high-speed DAC, which brings the digital signal into the analog domain. The finite resolution of the DAC also introduces spurious elements into the output waveform.

The value of the output frequency of a DDS system can be calculated as follows:

$$F_O = \frac{K * F_c}{2^N} \qquad (1)$$

In this equation, F_O represents the output frequency of the DDS, K is the static control word, F_c is the clock frequency, and N is the width of the accumulator register.

The magnitude of the frequency control word, K, can only take on integer values. Therefore the minimum frequency increment of the DDS system can be calculated with the following equation:

$$F_{inc} = \frac{F_c}{2^N} \qquad (2)$$

Additionally, this equation gives the value of the lowest output frequency of the system, as K is effectively 1.

2.7 Prior Work on Direct Digital Synthesis

The practice of Direct Digital Synthesis has been well documented since the 1970's – during that era its use was mostly confined to audio frequency synthesis, however, due to limitations in available technology [20]. Even in the early days of this practice, it was well understood and documented that spectral impurities existed at the output of the DDS due to discrete phase accumulation, phase truncation, etc. By 1981, steps were already being taken to stymie these spurious elements, and the first approach that was shown to work involved dithering the accumulation register by periodically adding a random number to it [21].

By the late 1980's, DDS technology had progressed such that outputs could now reach up to 100 MHz [22]. In 1987, the first year a device with this capability was reported, the die size for the DDS system was 4.65mm x 4.80mm [23]. This is massive by today's standards, but for the time, the extremely high speed of the device was cutting edge.

Clock rates in DDS systems are often limited by the adder chains which can become quite involved for high resolution systems. One way around this is to pipeline these adder chains. By the end of the 1980's, this was already being investigated in systems with resolutions up to 16 bits [24]. The designers of this pipelined system, which was fabricated in a 1um GaAs process, suggested that clock rates of nearly 1 GHz would be possible [24]. Additional documentation for pipelined DDS systems can be found in [25], which was used for a radiation sensitive application.

By the end of the 2000's, clock speeds well into the GHz range were possible in 24-bit systems that implemented pipelined adder chains [25]. These high clock rates were also made

possible through decreases in process size, as this particular design was fabricated in a 0.13um SiGe process [25]. Most recently, DDS techniques have been applied to wireless charging systems [26]. The use for these architectures is seemingly endless – one of many reasons why this approach has been used continuously for almost 50 years.

2.8 Description of the Fully Realized Digital Designs

When work on the K-State Ag-Radar IC commenced, the research team was working in a 180nm process. At the request of our sponsors, a shift was made to the 45nm process. There was high confidence that the overall size of the design would decrease drastically, and that the digital designs would operate at much higher clock rates than previously possible in the larger process.

Unfortunately, due to the legal red tape that comes with IC processes, the K-State Ag-Radar team has yet to gain access to the 'back-end views' of the digital library in the 45nm process – thus work was only able to be completed using their analog RF libraries. The 'back-end views' allow for layout synthesis. The research team did have access to 'front-end views' which allowed for netlist synthesis, but without the ability to synthesize layouts, work stalled on the digital block.

It was believed, prior to making the process change, that mixed signal synthesis and simulation (the marriage of the digital designs and the digital-to-analog converter into one 'block') would be the fundamental topic investigated in this document. Obviously, plans were adapted to meet the circumstances, and the document now focuses mainly on the digital-to-analog converter. That said, the digital designs were described in Verilog, simulated rigorously, and verified using FPGA development boards prior to work being done on the DAC. For that reason, a few pages describing the specific digital designs in detail are certainly warranted.

2.8.1 Control Registers

As stated earlier, the key function of the digital block is to generate a sinewave via direct digital synthesis. It requires very few lines of Verilog code to describe a basic DDS that outputs a static frequency from a given control word. The design in question, however, is a bit more complex. Recall that the Ag-Radar IC is achieving stepped frequency functionality by sweeping the reference frequency at the input of the PLL. This requires a different mode of operation from the DDS – a mode in which the control word is increased incrementally after N number of clock cycles. Several other modes (all 24 bits in value) were described as well – one mode allowed for the control word to be decremented every N number of clock cycles. Both of these modes increment or decrement in a linear manner. Another mode allows for a ramping up and then down of the control word. Still another mode allows for an exponential ramp of the DDS output.

These modes of operation are selected by the user of the IC via an off-chip SPI master module which then transmits data to an SPI slave on-chip – this mode of operation corresponds to one of a number of predetermined values which are then stored in a configuration register.

Register Name
Mode Select
Phase Increment
Frequency Increment
Number of Frequency Steps
Dwell Time

Figure 8 – The configuration registers for the DDS

There are a number of other configuration registers besides the mode select. They are shown in Figure 8. Figure 9 provides graphical explanation of their function and purpose.

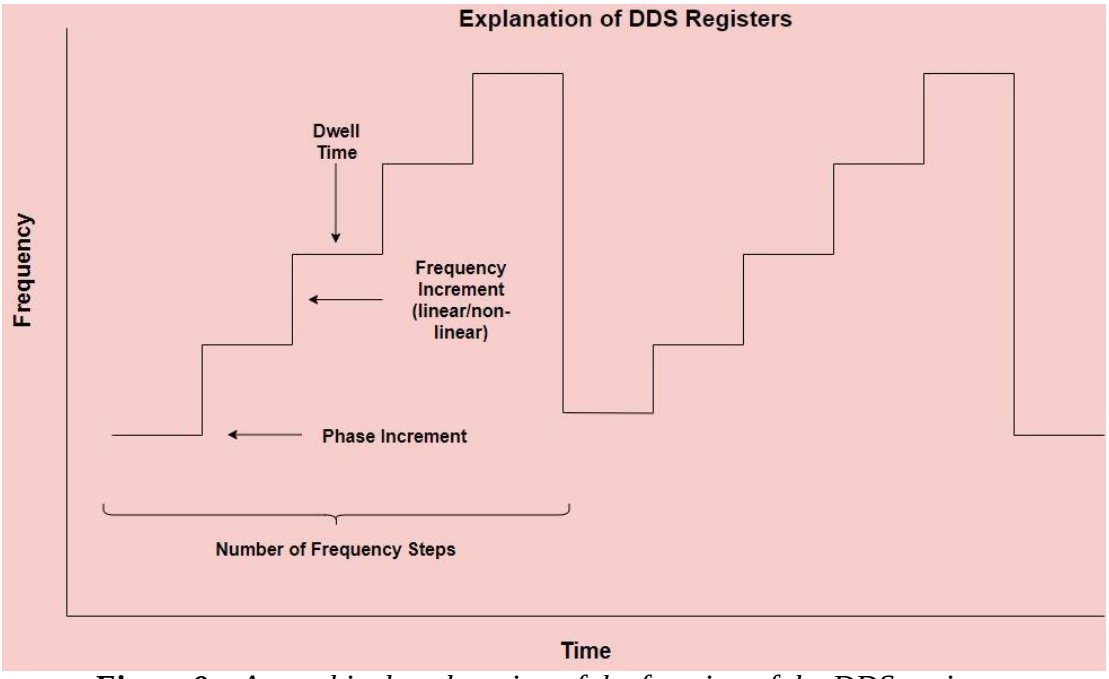


Figure 9 *– A graphical explanation of the function of the DDS registers*

The last aspect of the digital design is a memory block which serves as a lookup table. Several approaches were considered, and a number of them were tested in a laboratory environment. The first approach involved simply hardcoding a lookup table into a Verilog file using a MATLAB function which was developed by Dr. Gruenbacher. This approach was certainly the most streamlined, as it provided little to no hassle to implement. The problem of course, is that hardcoding the lookup table provides minimal versatility, while requiring a fair amount of chip-area during layout synthesis. A significant amount of combinational logic comes into play when hardcoding a lookup table.

Another approach is to synthesize a block of ROM. This would save chip-area, and allow exploration of memory options provided to us by the IC process. The ROM approach proved to be easy to implement in a laboratory environment on a Terasic DE2-115 board using IP from Intel, but much more difficult to implement on chip. Several attempts were made to contact the vendor of the 180nm process as we were unable to synthesize the on-chip ROM in house.

Eventually, the team ended up changing processes. Further investigation of the ROM block was shelved as the transition was made from one process to another.

More recently, there have been discussions regarding the implementation of a RAM block in the 45nm process. This has several advantages. First the DDS would no longer be limited to sinewaves. Any standard function, as well as completely arbitrary waveforms could be loaded into the RAM. This approach has some appeal, even if it may have little application in the agricultural field. Secondly, the ability to remap the memory block on demand could allow for subtle alterations of the data encoded in the RAM such that nonlinearities in the DAC could be calibrated out, or at least made less severe. The utility and of this will become more clear when the actual chip is fabricated and the research team is able to test its performance. There will be further discussion of this concept in a later chapter.

2.8.2 Verification Methods

Once the designs had been described and simulated, they were tested in a laboratory environment using the DE2-115 development board (the same development board from Terasic which was mentioned previously) - housed within was a Cyclone IV E FPGA. This development board also had a 50 MHz master clock, which was routed to the synchronous portions of the digital design during synthesis. This is a crucial part of the story in the development of this design. In the early stages of this project, it was believed that the target output range for the DDS would be approximately 10 MHz. In this case, a master clock frequency of 50 MHz was optimal – it provided a light oversampling rate of 5:1. This master clock frequency was relatively low; low enough that timing concerns should be almost non-existent, even with 24-bit adder chains.

This was then the perfect scenario – testing of the design in lab would mimic very closely the actual operating conditions of the chip. This soon changed, however, as system needs pushed

us to target a higher DDS output frequency – 10x higher in fact – 100 MHz. This of course corresponded to a 10-fold increase in the master clock frequency bringing it up to 500 MHz. Lab testing, while still useful to verify the design's functionality, became less of a priority at this stage due to timing issues present on the board at a 500 MHz clock rate. Synthesis and analysis via the tools for netlist synthesis and layout synthesis became most essential.

The netlist synthesis tool was able to provide the research team with timing reports, which allowed us to better assess how to proceed. There is a significant learning curve to any unfamiliar toolset, and this proved no exception. Configuring the digital design flow proved to be an iterative process (and one that will certainly go through further iterations when Kansas State is able to gain access to the 45nm digital libraries), and even seemingly simple tasks such as interpreting the data from a netlist-synthesis report was initially a challenge.

It was quickly discovered that clocking the digital block at 500 MHz was pushing the limits of the 180nm processes' speed capabilities for this given application. With a few minor changes to the adder chain configuration in the Verilog code, along with some adjustments to the scripts used to execute the digital flow, timing was met at 500 MHz – but with little to no room to go beyond that. Some more advanced techniques such as pipelining the adder chains were suggested, and explored in simulation with good results – in fact, initial timing tests on a pipelined 24-bit adder chain indicated that a clock frequency greater than 1 GHz was possible at the 180nm node. It was about this time, however, that the transition was made from 180nm to 45nm so further implementation of pipelining was not necessary.

Before the digital design work was completely halted, the iteration of the design that met timing at 500 MHz was given a 'final' pass through the design tools. In addition, each individual submodule of the digital design was passed through the netlist synthesis tool. A layout synthesis

was also executed to provide a more thorough analysis of timing, etc. An image containing timing and area information is shown in Figure 10. The layout is shown in Figure 11.

Digital Submodule	Fmax	Area (gates)
SPI Slave	700 MHz	19697
Phase Accumulator	500 MHz	36091
ROM	-	522
Complete Digital	500 MHz	59601

Figure 10 *– Timing and area data for the digital design in the 180nm process*

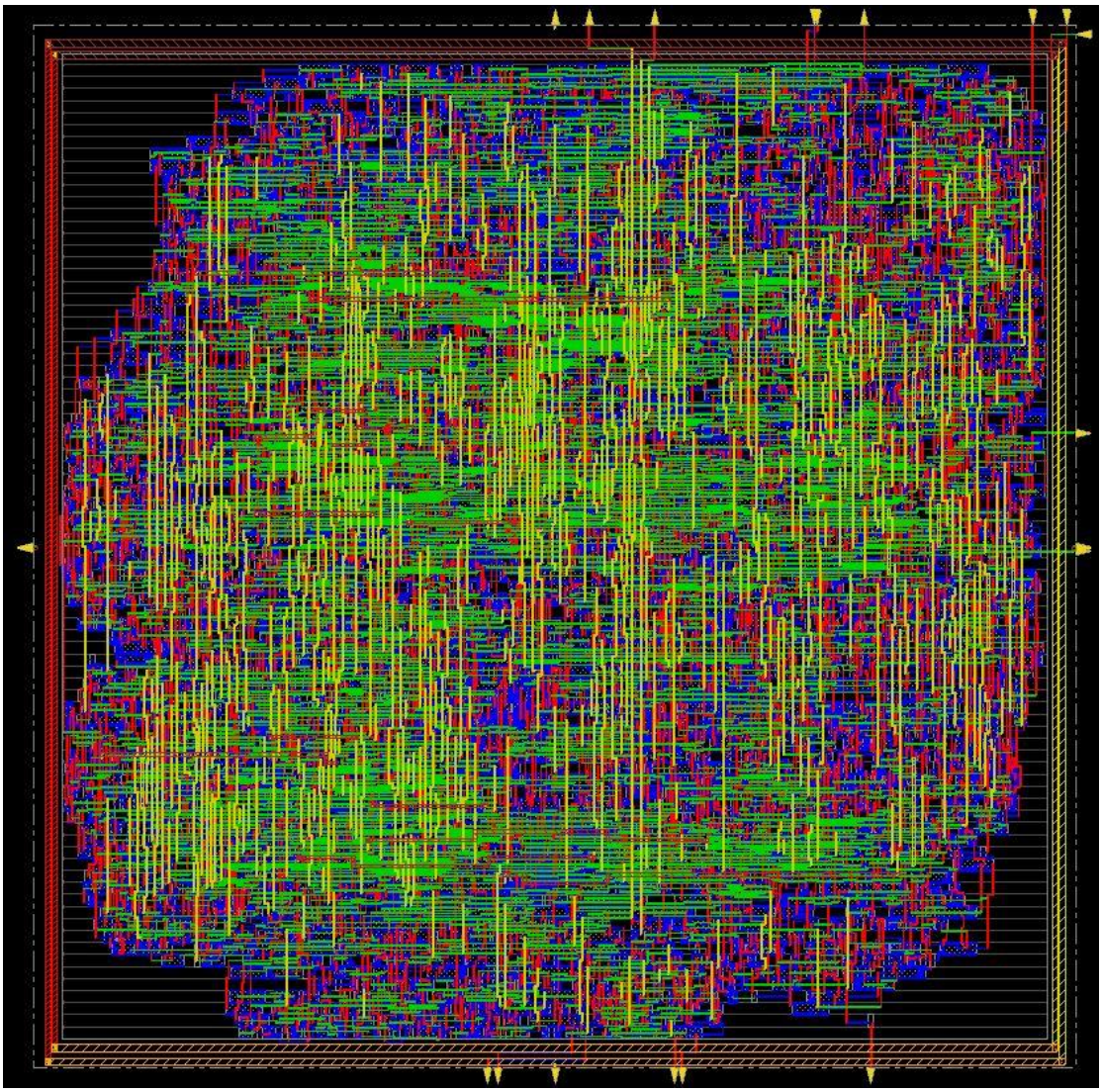


Figure 11 *– The layout of the digital designs in the 180nm process*

In the time since the research team has gained access to the 45nm process, the digital designs were synthesized (remember the 'front-end views' from earlier discussion). Of course, this was merely a netlist synthesis. However, some data was able to be extracted from this run. The figure on the following page displays some of that data. Note that data was gathered on the potential size of a RAM block.

Digital Module(s)	Sequential Cells	Logic & Inverter/Buffer Cells	Area (um^2) (based on cell size only)
SPI Slave	212	766	14,194
Phase accumulator, frequency accumulator & sweep control	148	2349	30,757
Phase accumulator (only)	24	296	4,199
RAM(256x12)	3072	19,335	399,216

Figure 12 *– Layout information for the digital designs in the 45nm process*

Chapter 3 - Digital-to-Analog Converter Topologies

3.1 Overview of the DAC

Digital-to-analog converters can be implemented in a variety of ways, with each architecture having advantages and disadvantages. Additionally, the application of the DAC is seemingly limitless in today's technological environment, where low-power, high-speed, and extreme precision are paramount to the performance of cutting-edge electronics. One can expect to find an integrated circuit (IC) chip containing a digital-to-analog converter in nearly every cellular device, television, and audio playback system on the market today, as well as many in radars used both commercially and in the defense sector.

Digital-to-analog converters work by taking a digital input word and outputting a voltage or current in the analog domain. For each possible digital input, a corresponding analog value exists. Since finite-length digital words have limited precision, this creates an output that is akin to steps in a staircase when a sequence of digital words is processed by a DAC. The smallest step between two analog values (e.g. voltages) that a DAC can resolve is referred to as '1 LSB', meaning of course that if the LSB of the incoming digital word is flipped relative to the previous (while all other bits remain the same), the '1 LSB' delta in the analog voltage will result. This minimum analog voltage delta, defined as 'R' in Equation 3 below, can be easily calculated:

$$R = \frac{Vdd}{2^N} \qquad (3)$$

Here, Vdd is the supply voltage of the DAC, and N is the number of bits (resolution) that the DAC takes as an input.

Clearly an analog output waveform resembling a staircase is not ideal in practical applications. For this reason, a reconstruction filter (lowpass) is almost always present at the

output of a DAC in order to "smooth" the waveform by removing sharp transitions between the analog voltage levels which contain exceedingly high frequency components.

DACs are almost always implemented in an IC due to the component matching requirements of modern systems, as well as obvious sizing constraints. As modern IC fabrication processes continue to delve deeper into the sub-micron realm, DACs continue to get smaller and faster, and can boast higher resolutions. Resolution, however, is just one of the performance metrics by which digital-to-analog converters are characterized. Others include *Maximum Sampling Rate* – the greatest speed at which a DAC can produce a correct output, *Dynamic Range* – the difference between the largest and smallest signals the DAC can reproduce, *Monotonicity* – the property by which a DACs analog output only increases as the digital code increases and vice-versa. Another figure of merit is *Integral Nonlinearity (INL)* – the deviation between the ideal output value of the DAC and its actual measured output. This can be calculated as follows, where c is the input code to the DAC, and N_c is the output code:

$$INL = \left|\left(V_{O,max} - V_{O,min}\right) * \frac{c}{N_c - 1} - V_{O,measured}(c)\right| \qquad (4)$$

Another is *Differential Nonlinearity (DNL)* – the deviation between two analog values corresponding to adjacent input digital values. The equation for this figure of merit is:

$$DNL(i) = \frac{V_{out}(i+1) - V_{out}(i)}{ideal\ LSB\ step} - 1 \qquad (5)$$

Still another is *Spurious-Free Dynamic Range (SFDR)* – the ratio between the powers of the desired signal and the undesired frequency spur of greatest magnitude. A final figure of merit to note is *Total Harmonic Distortion* – a metric quantifying the noise introduced in the analog output signal by the DAC.

3.2 Resistor-Based DACs

Implementing a DAC with resistors is generally regarded to be the simplest approach. Resistor-based DACs come in several varieties: ladder, binary-weighted, and R-2R. Ladder DACs consists of resistors placed in series, such that they divide a reference voltage into an appropriate output voltage. Between each resistor is a tap, connected to a switch which selects whether the given resistor will be included in the voltage division process. This is highlighted in Figure 13. The number of resistors in the series array can be found by taking 2^N where N is the bit-resolution of the DAC. Therefore, this topology also requires binary-to-thermometer decoders (not shown in the figure), which consume more chip area.

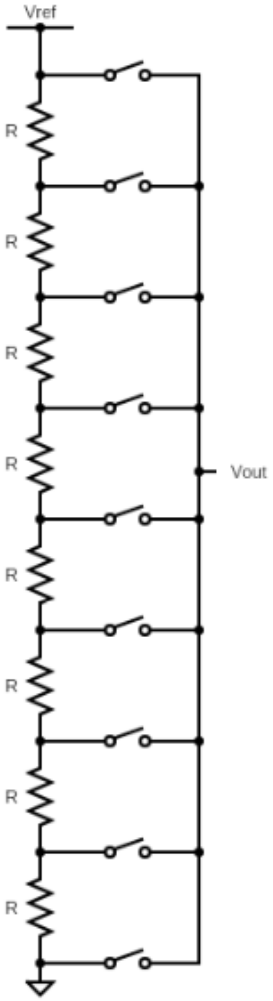


Figure 13 *– A basic 3-bit ladder DAC*

Ladder DACs are monotonic, and very simple to conceive in a theoretical design environment. One obvious disadvantage, however, is the large number of resistors required for a high-resolution system. This topology also introduces significant layout challenges, as resistors commonly take up a large amount of area on an integrated circuit. Additionally, matching a large number of resistors is not guaranteed in modern IC processes – this of course causes linearity issues which can severely hamper the reliability of the DAC.

Another common resistor-based topology is built upon a parallel array of binary-weighted resistors. In practice, this topology almost always includes an op-amp, as well, which serves as a buffer to pass the analog voltage generated by the current flow through the resistors to the output of the system. A basic three-bit example of this is shown in Figure 14.

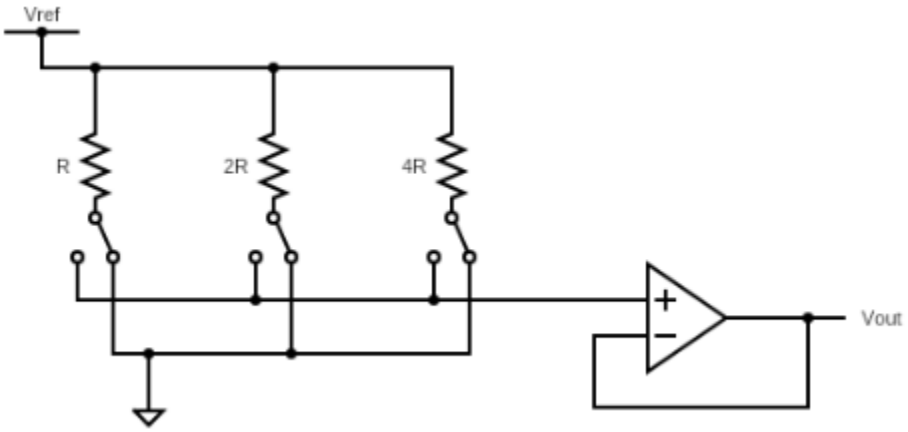


Figure 14 *– A binary weighted resistor DAC*

This topology suffers from many of the same drawbacks as the ladder-DAC. To further emphasize this, take the example of a 10-bit DAC, a resolution commonly seen in modern implementation. The resistor representing the MSB of the system will be 1024 times larger than the resistors representing the LSB. If the tolerance of the MSB resistor is less than 0.1%, nonlinearities of greater than 1 LSB will result. If tolerances grow larger than 1%, relatively

large errors can occur. Such errors are usually unacceptable in practice, and it's no surprise that this topology is found more often in textbooks than in fully realized IC chips.

The last resistor-based architecture to be explored is the R-2R DAC. This is similar in topology to the binary-weighted DAC discussed previously, except that it only uses two resistor values to perform voltage division. This topology requires 2N resistors – depending upon the resolution this could be an improvement or degradation in chip-area consumed, as size of resistors in an IC process is directly correlated with the desired resistance value. Figure 15 shows the R-2R DAC from a schematic level view.

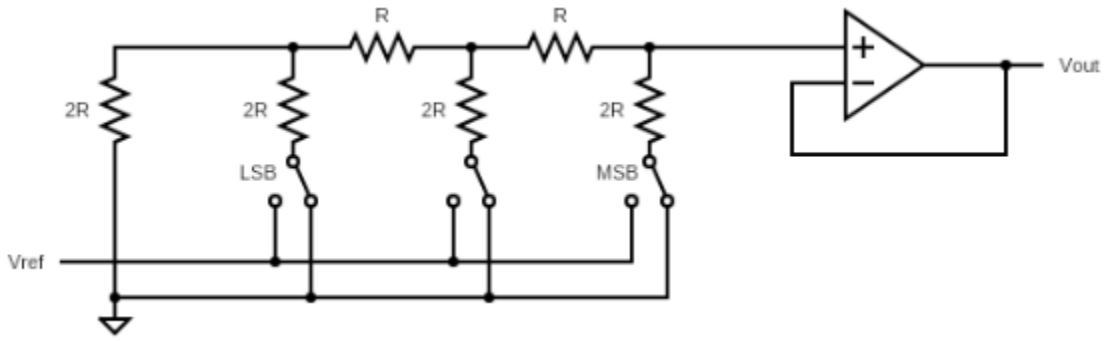


Figure 15 *– An R-2R DAC*

This is probably the most commonly implemented resistor-based topology implemented in practice. It should be noted, however, that it's commonly used in tandem with current source transistors in a sort-of hybrid topology to achieve higher resolutions.

3.3 Charge-Scaling DACs

Charge-scaling DACs have topologies much the same as resistor-based DACs with the difference being, of course, that capacitors are used in place of resistors. It follows then, that a charge-scaling approach could be implemented in a ladder architecture, as described earlier for a resistor-based DAC, or in a binary weighted, or C-2C topology. In this section, a ladder topology

and a binary-weighted topology will be described in some detail, with schematic view included. A capacitor based ladder DAC is shown in Figure 16.

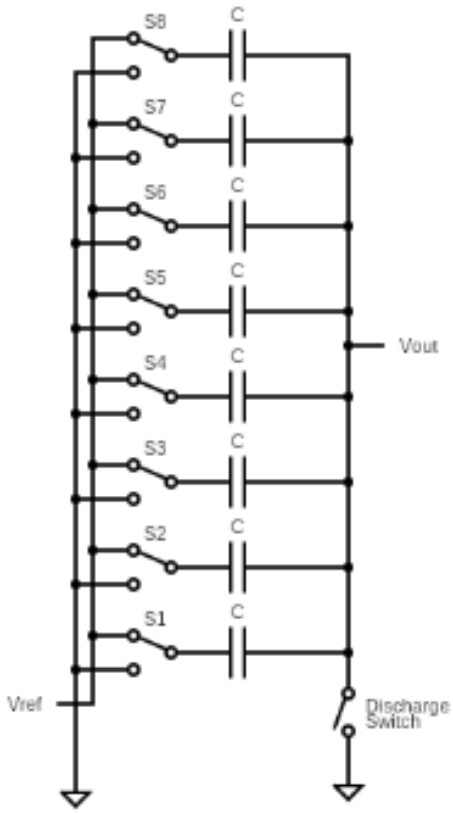


Figure 16 *– A charge scaling DAC with a 'ladder' architecture*

In this topology, 2^N capacitors are arrayed in parallel, and binary-to-thermometer decoders drive S1 – S8, which either connect to ground or to the reference node. When a switch is connected to the reference, charge, calculated using Eq. 5 below.

$$Q = C * Vref \qquad (6)$$

The amount of charge, Q, is injected onto the output node – meanwhile the discharge switch is open. Then when the discharge switch, at the bottom right of the circuit in Figure 16, closes, the capacitor array is discharged. For this charge-scaling ladder-DAC scheme to function properly, more intricate switching is required (when compared to a resistor-based ladder DAC),

and other timing constraints must be characterized and correctly accounted for to ensure proper functionality.

Binary weighted charge-scaling DACs come with an interesting catch, when compared with their resistor-based counterparts. The catch is this: implementing the higher order bits by simply multiplying the capacitance value by the binary weight is impractical. Large capacitance values require extreme amounts of chip-area, and can make such a simple design approach impractical. The solution is to insert an attenuation capacitor between the 'LSB array' and 'MSB array' such that the C values of the MSB array and LSB array are equivalent in weight. This is shown in Figure 17.

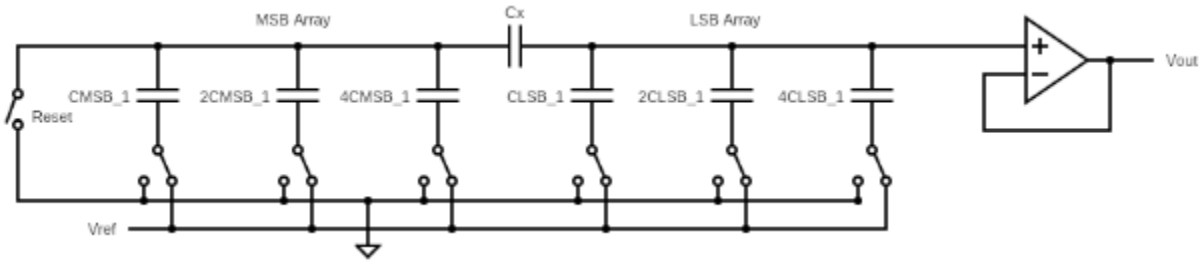


Figure 17 *– A charge scaling DAC with an MSB and LSB array, separated by an attenuation capacitor*

If one denotes the total capacitance of the LSB array as C_{LSB}, and the first capacitor of the MSB array as C_{MSB_1}, then the value of the attenuation capacitor C_x can be calculated using Eq. 6.

$$\frac{1}{C_{MSB\_1}} = \frac{1}{\frac{1}{C_x} + \frac{1}{C_{LSB}}} \qquad (7)$$

This topology is implemented with relative ease in modern IC processes, provided the capacitor sizes can be kept to reasonable values. The drawback of this topology is that it requires an op-

amp to be designed on chip which can limit bandwidth, and of course, takes up valuable chip-area. Modern approaches to charge-scaling DAC design are discussed at length in [27].

3.4 Current-Steering DACs

Current steering architecture is arguably the most common, and best performing DAC architecture in use today. They are very high speed, high bandwidth, and consume a predictable amount of area, which of course, depends upon the topology used. Current steering DACs work by summing together currents sourced by parallel current-mirrors. The current-mirrors source current at all times, and a differential pair of PFETs are used to steer current to the output node, or to ground. The output node into which the desired current is steered contains a resistor, and the current flow across the resistor generates the analog voltage at the output of the system. The three most common topologies include binary-weighted, thermometer-coded, and partially-segmented.

Binary weighted DACs contain N number of current sources, and a schematic of said topology is shown in Figure 18 for three-bit resolution.

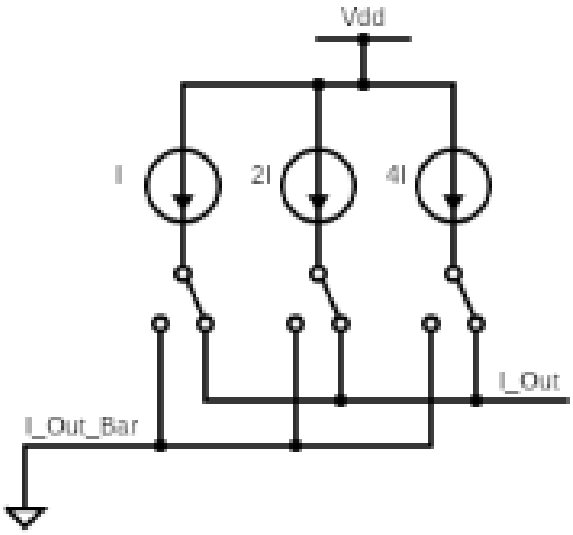


Figure 18 *– A 3-bit current steering DAC*

This topology requires no additional decoding logic, consumes relatively low amounts of chip-area, and can be implemented with a small number of current mirrors. That said, purely binary weighted DACs are rarely seen in practical implementation. This is due to the fact that they are non-monotonic, which can contribute to major glitches and other non-idealities when multiple current mirrors are being switched at the same time.

Another approach, thermometer-coding, corrects many of the problems present in the binary-weighted topology. In a thermometer coded, or fully segmented topology, each current cell flows '1 LSB' of current. A numerical explanation of thermometer coding can be seen in Figure 1. Note that the example shown is for three-bit resolution.

Binary Word	Thermometer Code
000	0000000
001	0000001
010	0000011
011	0000111
100	0001111
101	0011111
110	0111111
111	1111111

Table 1 *– Binary to thermometer coding*

Thermometer coding requires 2^{N-1} current cells to exhibit proper function, which certainly consumes more area than binary coding, but also reduces glitches upon code transitions, due to its monotonic nature. Since each current cell flows 1 LSB of current, it also allows for smaller FETs to be used throughout the system, which further aids in mitigating glitches upon code transitions due to lower parasitic capacitances. A three bit binary coded DAC is shown in Figure 19.

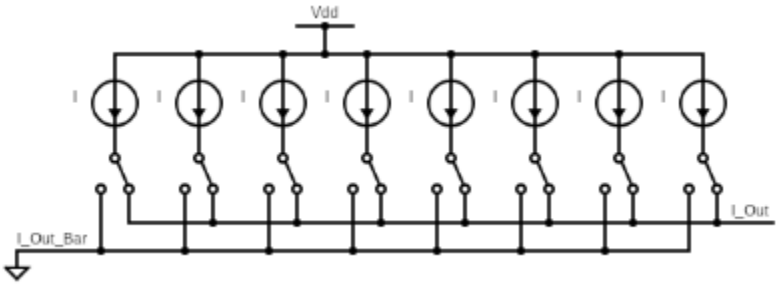


Figure 19 *– A 3-bit thermometer coded current steering DAC*

It is common to array the current cells for a fully segmented topology in a square matrix. Figure 20 demonstrates how a designer might configure the manner in which the array 'fills-up' as the incident digital word increases in size, thus turning on more current cells.

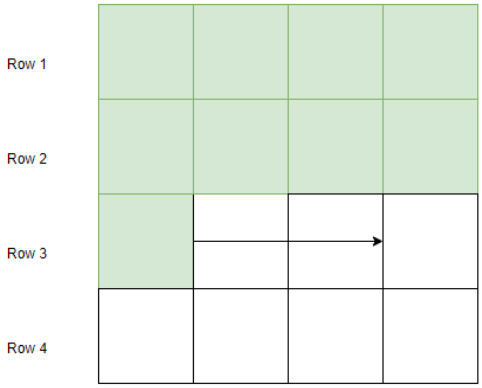


Figure 20 *– A diagram demonstrating the manner in which the current-cell matrix is engaged*

Chapter 4 - 12-bit Current Steering DAC Design

4.1 Overview

As the previous chapter indicated, there are many options at the disposal of an electronics designer seeking to implement a digital-to-analog converter. That said, the current steering topology was undoubtedly the best choice for the K-State Ag-Radar. The high-speed capabilities of the architecture, as well as its low-power consumption and total area in the 45nm CMOS process allowed for 100% segmentation (full-thermometer coding). Conveniently, this also ensures high linearity and the lowest possible glitch energy upon bit transitions. The figure below shows a basic diagram of the system which was implemented.

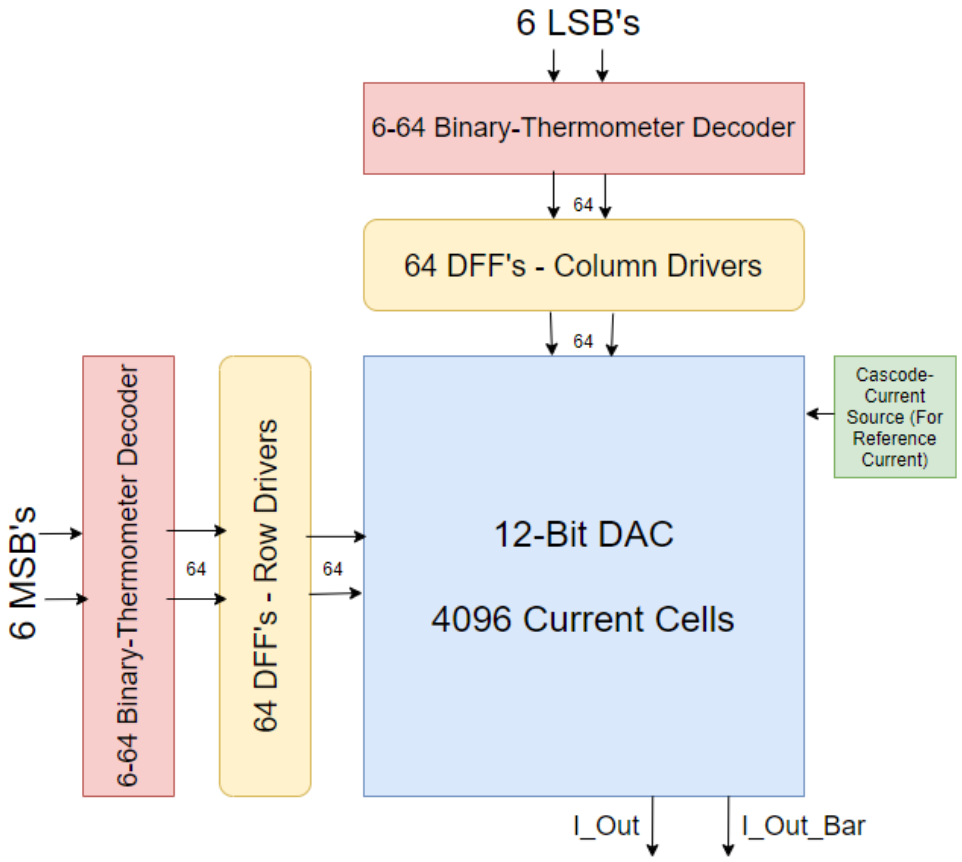


Figure 21 – A block diagram of the 12-bit DAC and surrounding substructures

The incident 12-bit binary word is divided into two six-bit sub-words. The 6 LSB's drive the column inputs of the DAC, but first must undergo binary-to-thermometer decoding, and then pass through a register to be made synchronous. The same is true of the 6 MSB's, which will drive the row inputs. The total number of current sources in the 'on' state is of course determined by the value of the incident word. To take the two extreme cases as examples, if the input to the DAC is 111111111111, then $4095*I_{REF}$ will be steered to I_Out, and no current will flow to I_Out_Bar. Likewise, if 000000000000 arrives at the input, then no current will flow to I_Out, and $4095*I_{REF}$ will flow to I_Out_Bar.

4.2 Current Cell

Logically, the next thing to describe in detail would the individual current cells which are arrayed in the 64x64 grid. The current cells contain digital gates, an analog current source, and a differential switch. The digital logic takes active high inputs from the row and column signals, and these in turn drive the switch, which steers current to I_Out or I_Out_Bar. The switch is active low in the sense that a 0 will steer current to the desired output node. The current source was implemented with PFETs and is cascoded to improve linearity and output resistance. The following figure shows a basic schematic of the current cell.

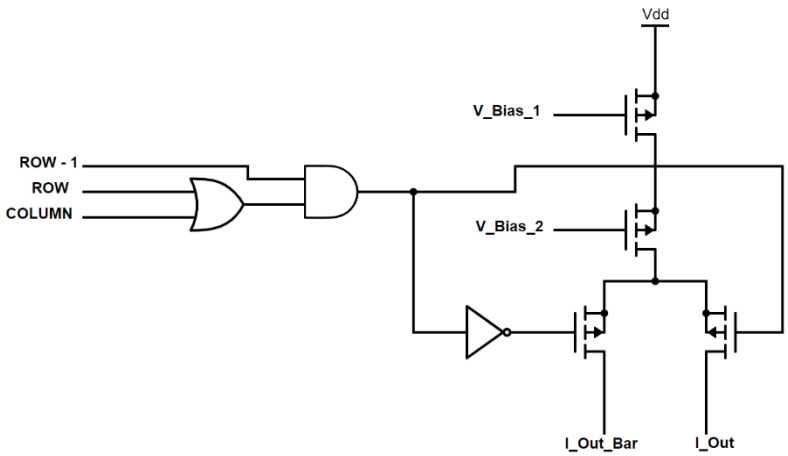


Figure 22 *– The schematic for an individual current cell within the DAC*

Each current cell was designed to flow approximately 7.5uA of current – this is the I_{REF} value which was discussed previously. Taking 4095*7.5uA yields about 30mA of total current flow in the 12-bit DAC. Figure 23 demonstrates the 'current-steering' mechanism of an individual current cell.

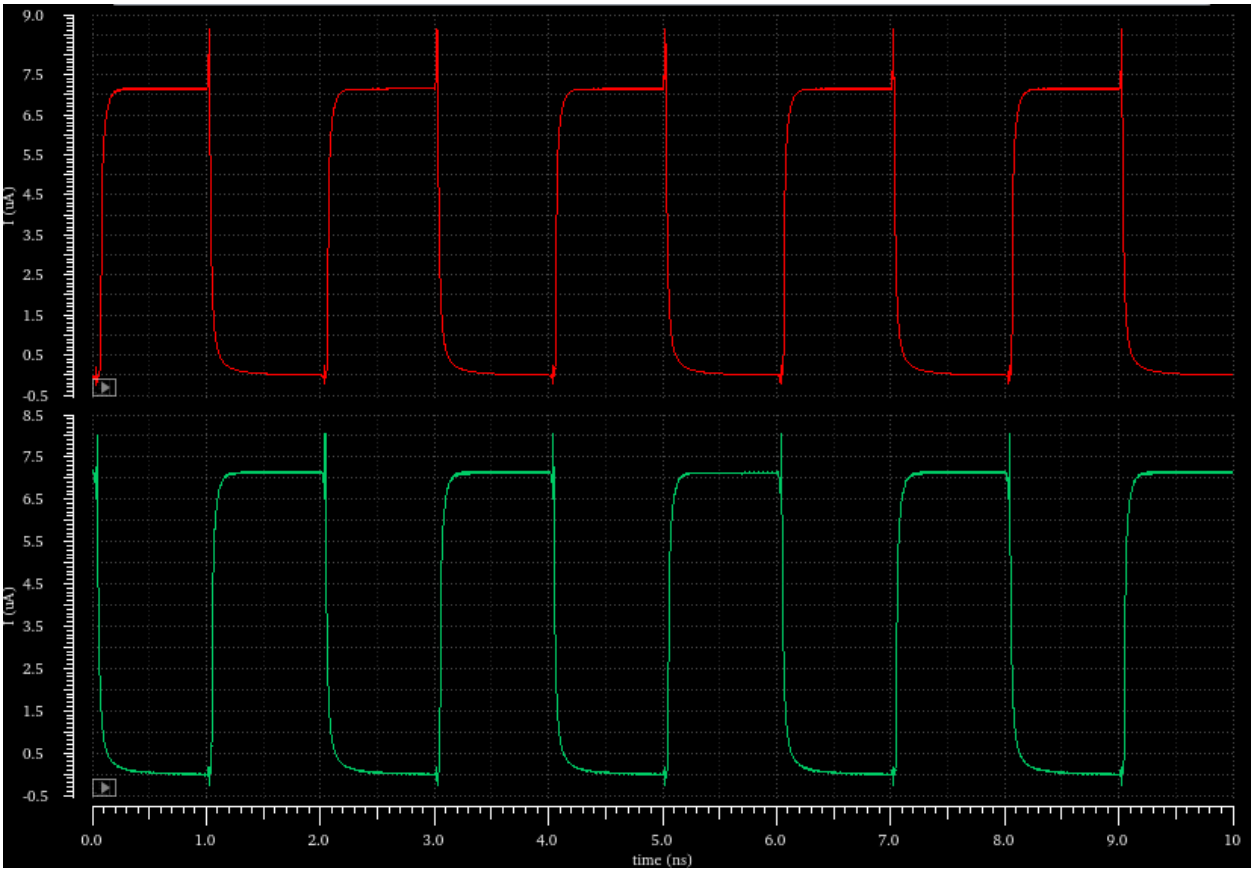


Figure 23 *– A plot of two complementary current waveforms highlighting 'current steering'*

Figure 23 shows two current waveforms – representative of the two complementary current outputs of the system. As current flows to one output, the other output goes to zero.

This individual current cell was then laid out. This can be seen in Figure 24.

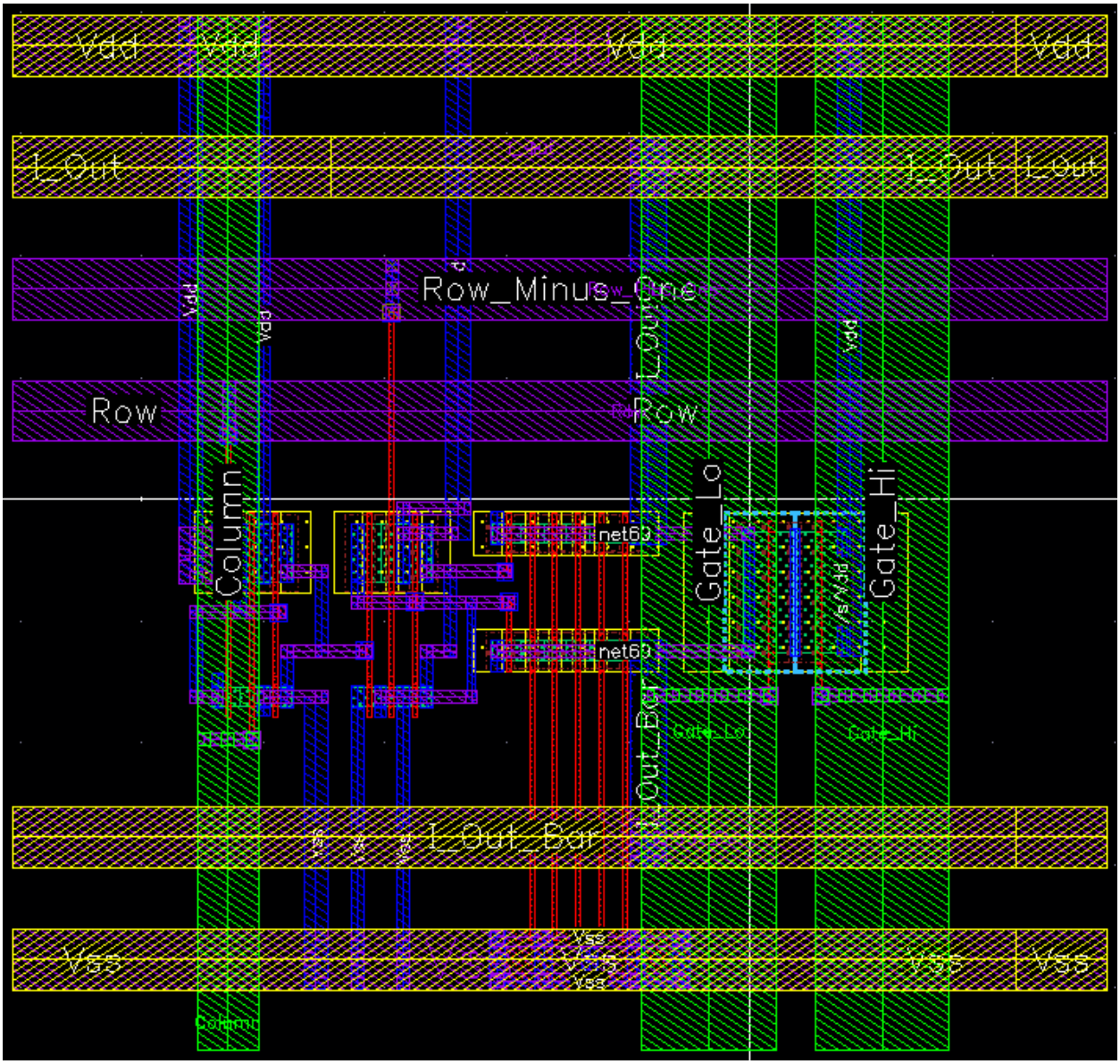


Figure 24 *– The layout of the individual current cell*

The dimension on the cell once everything was in place proved to be 9um in height by 8um in width. When this was fully arrayed it amounted to about 545um in height by 573um in width. In terms of area, this is 0.312 mm^2.

The two large FETs in the middle right of the image are the cascode pair used for the current mirror. The FETs in the middle left correspond to the digital logic ‘AND’, ‘OR’, and ‘NOT’ gates which were shown previously in the current cell schematic. The FETs shown in the middle of the current cell were not shown in the schematic, as they simply act as an on-chip RC filter to mitigate transient current spikes caused during bit transitions in the DAC. The PFETs used to achieve this filtering are ‘stacked’ in a faux-cascode topology onto the current steering pair of PFETs, and each PFET in the filter has its gate grounded. This puts the FETs in the triode region of operation where they act as small resistances. This filter was added after simulations were run on a previous iteration of the DAC that had no on-chip filtering. Figure 25 shows the prevalence of switching transients that comes without filtering as compared to a system with filtering whose plot is shown in Figure 26.

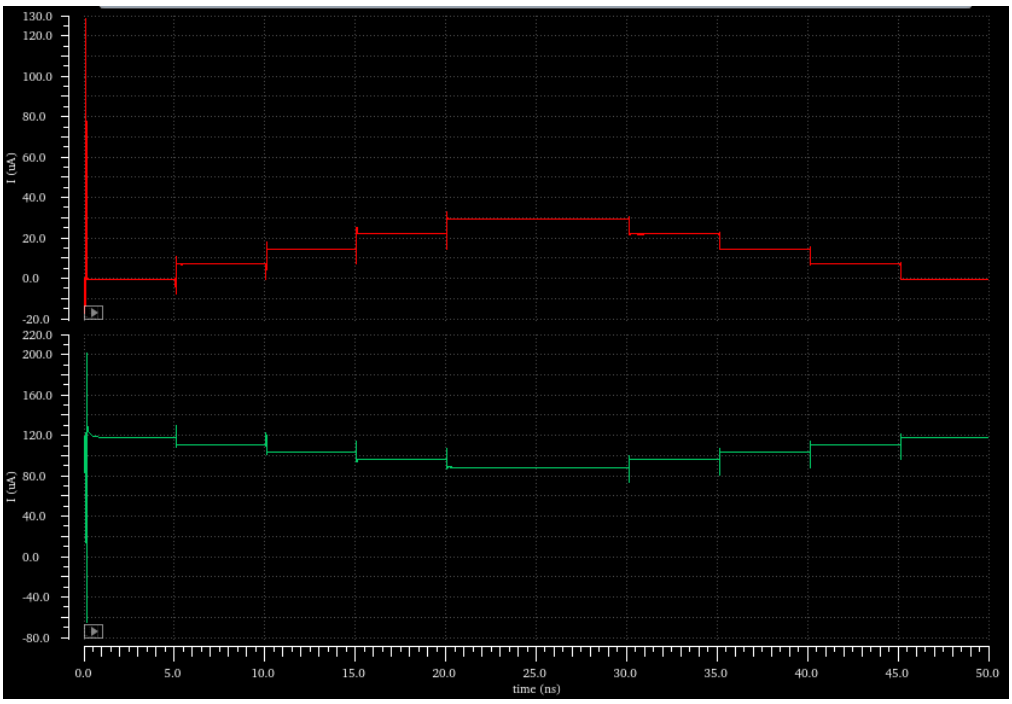


Figure 25 *– Complementary current waveforms with no on-chip filtering*

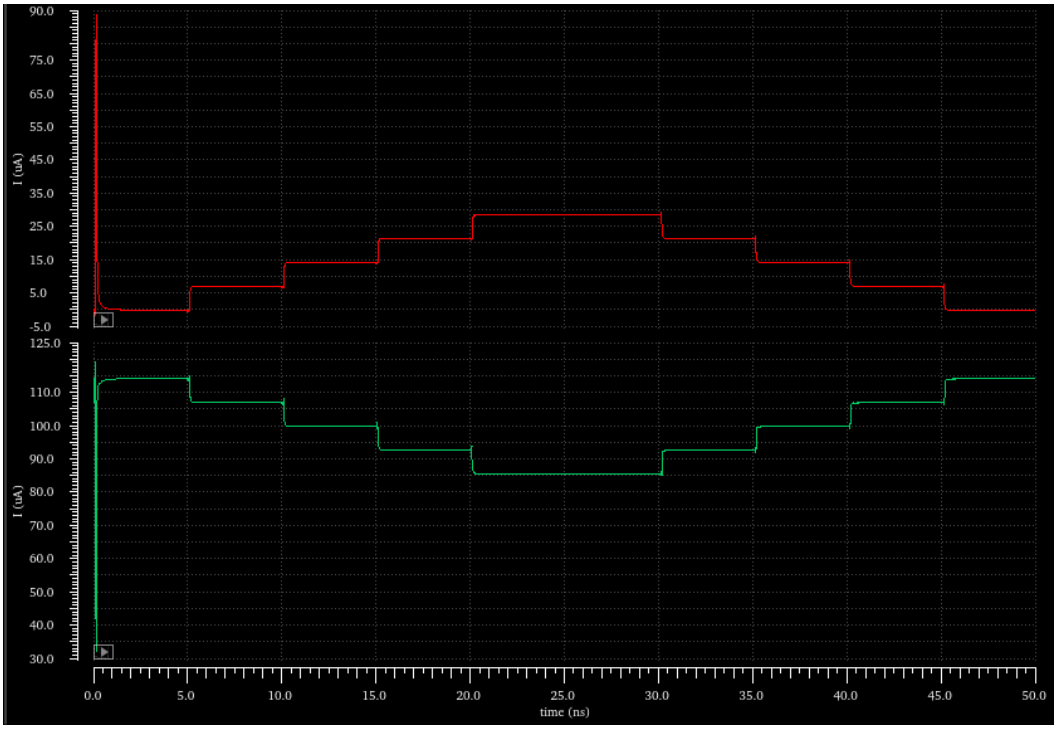


Figure 26 *– Complementary current outputs with on-chip filtering*

While the switching transients could have been dispensed with through filtering off-chip, attenuating their amplitude at the source proved to cost very little in terms of area, and thus the filter was included in the final design.

4.3 D-Flip-Flop

Moving on, let's examine the D-Flip-Flops that drive the row and column inputs of the DAC. The chosen topology used transmission gates as opposed to NAND gates, and the chosen schematic can be seen in figure. Each flip-flop needed the drive strength to turn 'on' or 'off' many inputs – 64 for the column driving flip-flops and 128 for the row flip-flops (by nature of the 'ROW' and 'ROW – 1' logic scheme used). Therefore, to ensure sufficient drive strength, an exponential horn buffer was placed at the output of the flip-flop, with sufficient strength to drive a 1pF load. This was certainly an extreme amount of robustness built into the design – far more

than needed, but there was plenty of chip area to work with, so the design proceeded with the large buffer.

The schematic can be seen in Figure 27, and layout of the DFF can be seen in Figure 28.

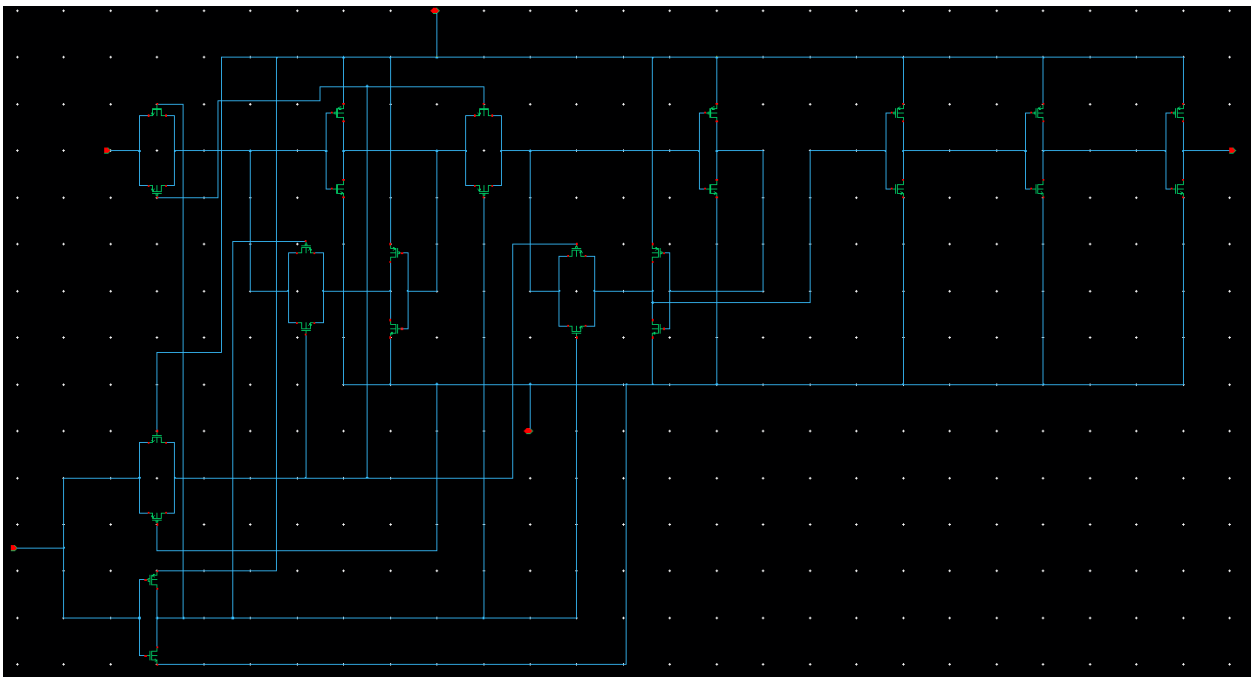


Figure 27 *– The schematic, based on transmission gates, used to implement a DFF*

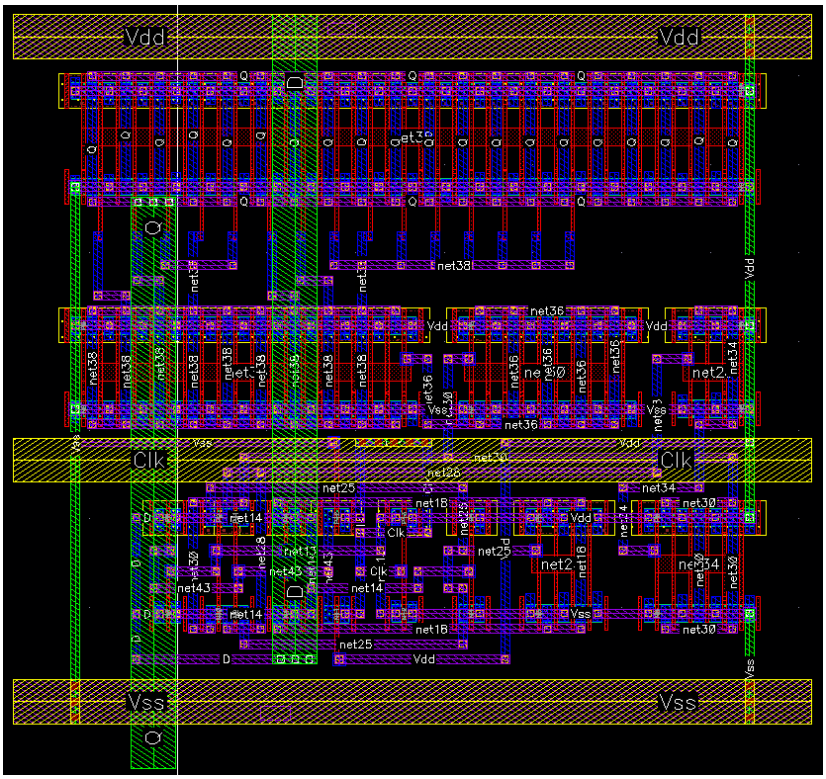


Figure 28 *– Layout of the DFF*

The actual D-Flip-Flop proper is located below the ‘Clk’ trace which runs through the middle of the layout. Above the ‘Clk’ trace is the large output buffer to step up the drive strength of the unit. This proved to be a very difficult design to layout due to the out of phase transmission gates – additional challenges came in trying to keep the DFF layout dimensions identical to those of the current-cells so that they could be abutted in the top level design such that no further ‘by-hand’ routing was required at that stage of the design’s hierarchy.

4.4 Thermometer Decoders

The next facet of the design includes the 6-64 binary-thermometer decoders. In a modern mixed signal design flow, these digital blocks would have been synthesized from an RTL design, with little to no ‘by-hand’ place and route from the designer. Due to factors outside of our

control, the same ones previously discussed in this document, the 6-64 decoders were done by hand using a design and verification flow more typical of analog synthesis.

Despite the fact that the designs couldn't be synthesized directly from an RTL script, they were still described in Verilog and appropriately simulated. The digital logic was 'simple' enough to comprehend from the Verilog, that a schematic could be recreated in the analog design tool that mimicked the functionality described in Verilog. Once these schematics were created and verified, the layout stage came next.

The layout proved to be very challenging, and this was compounded by the fact that two separate variations were required (one each to drive the rows and columns). To ensure that the functional integrity of the design was not compromised by human error, the 6-64 decoder was broken down into three levels of hierarchy - the first being a simple 2-4 decoder.

4.5 2-4 Thermometer Decoder

The schematic can be seen in Figure 29, and was drawn by Dr. Don Gruenbacher.

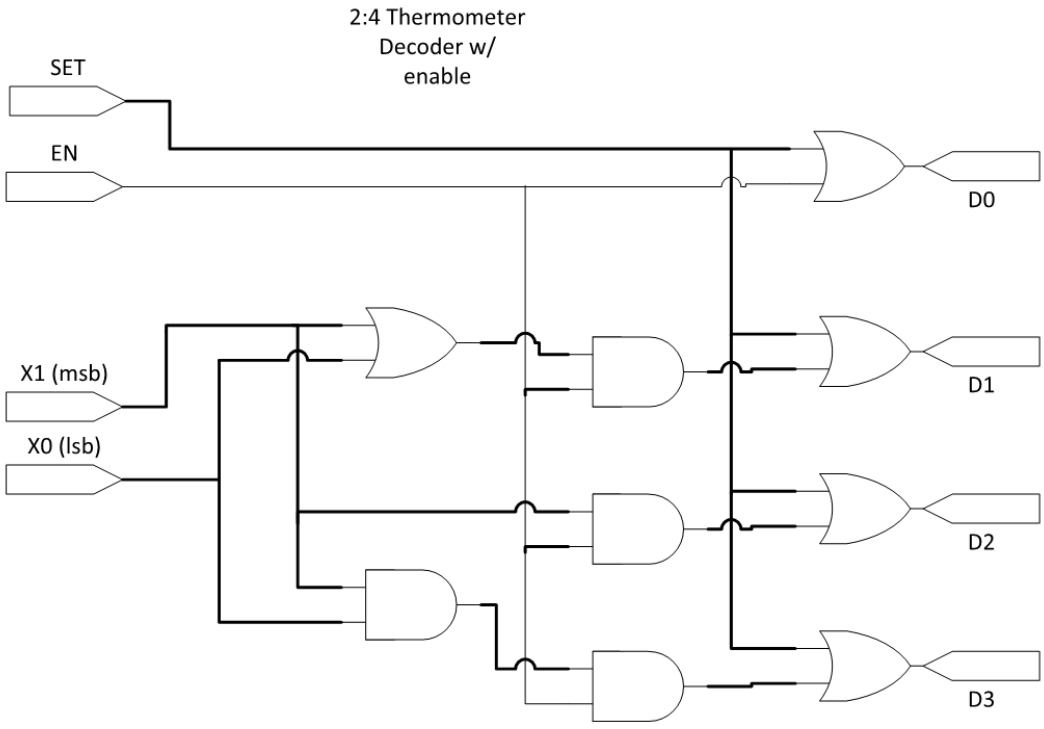


Figure 29 *– The schematic used to implement a 2-4 binary-thermometer decoder*

A layout for this cell was, of course, also completed. An example of a 2-4 decoder used to drive the column inputs of the DAC can be seen as follows.

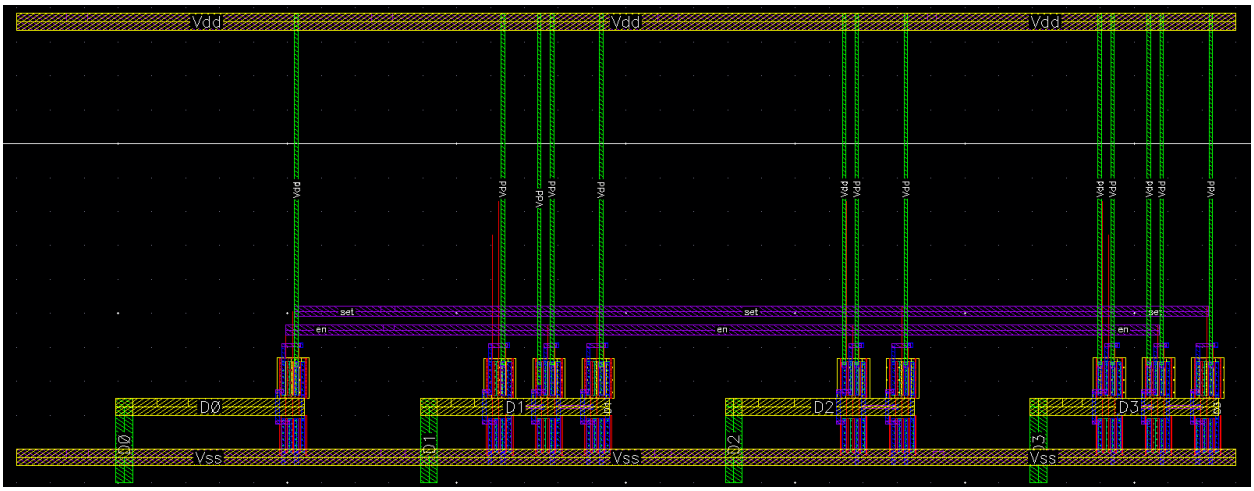


Figure 30 *– A layout of the column 2-4 decoder*

Once a cell was completed for the 2-4 decoder, the next level of hierarchy to be synthesized was a 4-16 decoder.

4.6 4-16 Decoder

The schematic can be seen in Figure 31, and was drawn by Dr. Don Gruenbacher.

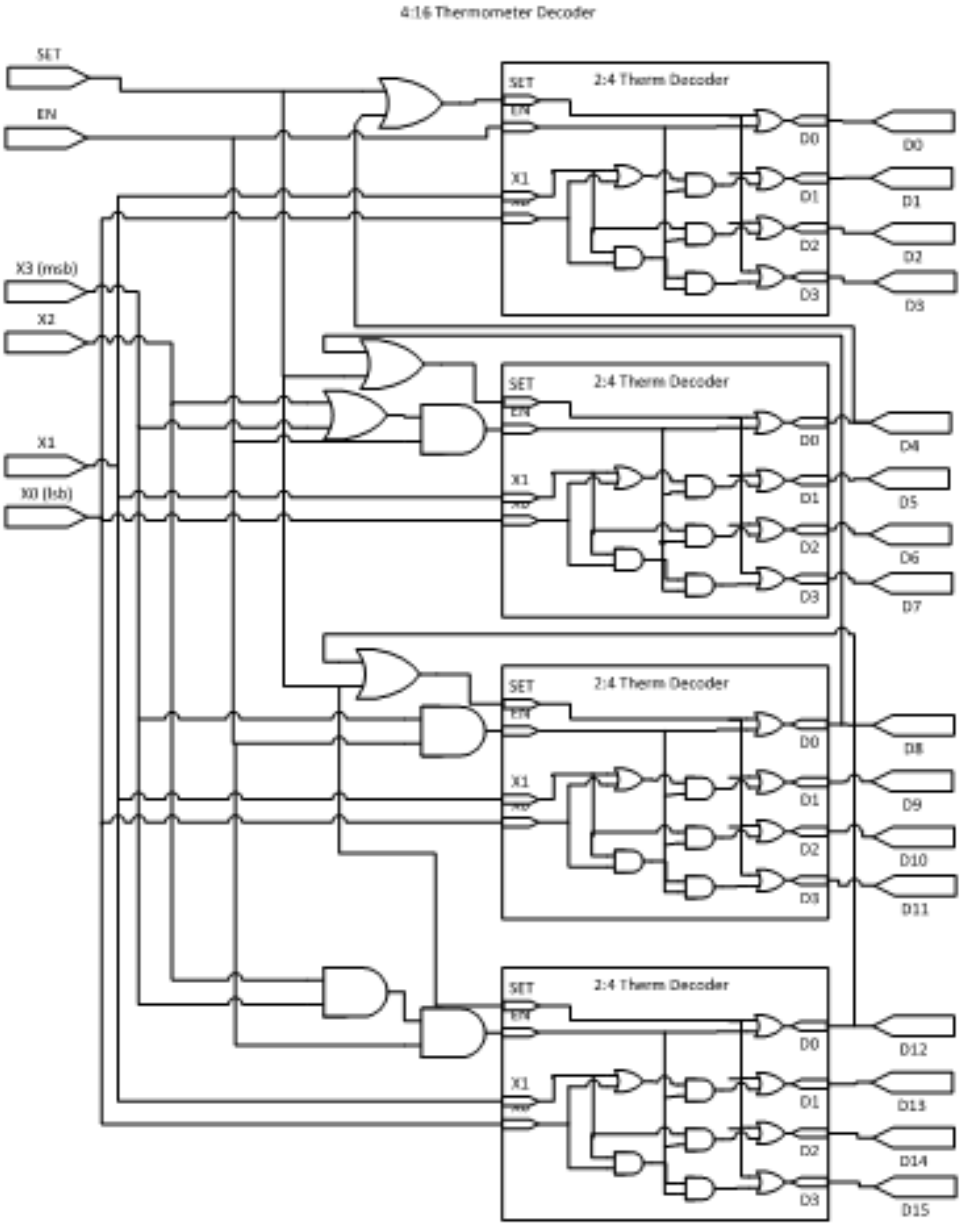
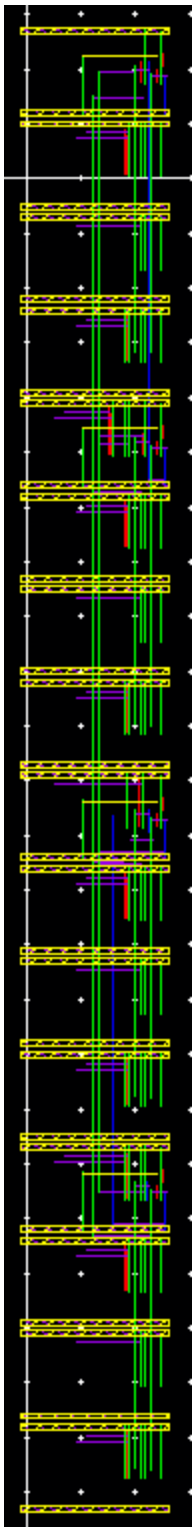


Figure 31 *– The schematic used to implement a 4-16 binary-thermometer decoder*

Note a small error in the schematic above, should an attempt be made by a reader to recreate this design – the line that seemingly connects D12 with D8 should be 'jumped' over D8 – thus they are not connected.

Though in a previous figure, an example of a column decoder was shown, at the 4-16 level of hierarchy, a row decoder will be shown for the sake of readability in this document. The 4-16 decoder to drive rows of the DAC can be seen in Figure 32a – due to resolution this is by no means able to convey much detail, but the overall structure of the layout can be discerned from the horizontal power and ground traces etc.



4.7 6-64 Decoder

At last, the 6-64 decoder could be fully realized. In lieu of showing schematics or layouts for this large design, the Verilog which Dr. Don Gruenbacher wrote to describe this circuit can be seen in Figure 32b.

```
module thermdec_6_64(x, d);
    input [5:0] x;
    output [63:0] d;


    wire    en0, en1, en2, en3;
    wire    set0, set1, set2, set3;

    assign en0 = 1'b1;
    assign en1 = (x[5] | x[4]);
    assign en2 = x[5];
    assign en3 = (x[5] & x[4]);

    assign set0 = d[16];
    assign set1 = d[32];
    assign set2 = d[48];
    assign set3 = 1'b0;

    thermdec_4_16 dec0 (x[3:0], en0, set0, d[15:0]);
    thermdec_4_16 dec1 (x[3:0], en1, set1, d[31:16]);
    thermdec_4_16 dec2 (x[3:0], en2, set2, d[47:32]);
    thermdec_4_16 dec3 (x[3:0], en3, set3, d[63:48]);


endmodule
```

Figure 32 (a) *– Layout of 4-16 binary-thermometer decoder* ***(b)*** – The Verilog description of the 6-64 decoder

4.8 Buffers

The last aspect of the physical design and layout is buffering, which improve the signal integrity of all digital inputs to the DAC. Input buffering is essential to assuring that digital signal lines present a light load to off-chip input signals while supplying fast rise-times to internal circuits for consistent operation up to 500 MHz. To emphasize the importance of buffering, recall that the input clock must drive 128 flip-flops.

All buffers consisted of four inverters in series. Different sizing and layout techniques were used depending on which input signal the buffer would be treating. The following image shows the layout for the clock buffer.

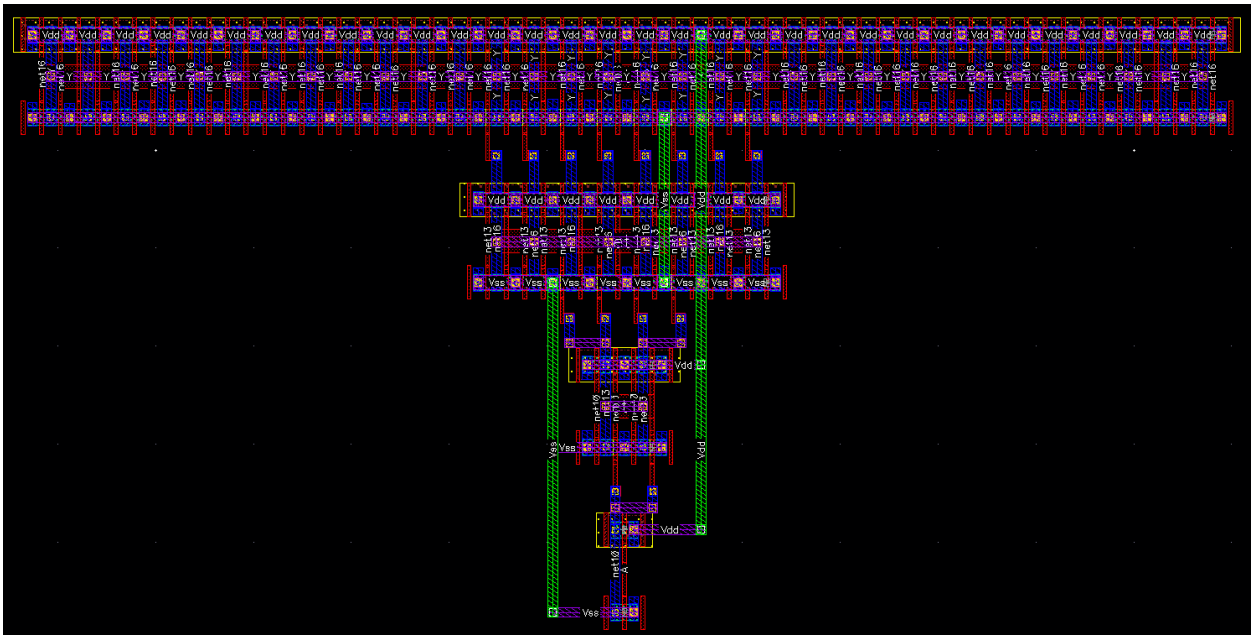


Figure 33 *– Layout of the clock buffer*

The number of fingers in each FET was manipulated such that all gate lengths were consistent which allowed for the ‘tornado’-like layout seen in Figure 33, and maximizes internal clock transition speed.

This approach was not taken, however, for the buffers which received the digital input word. An image of a buffer used for an input to a ‘column’ is shown next.

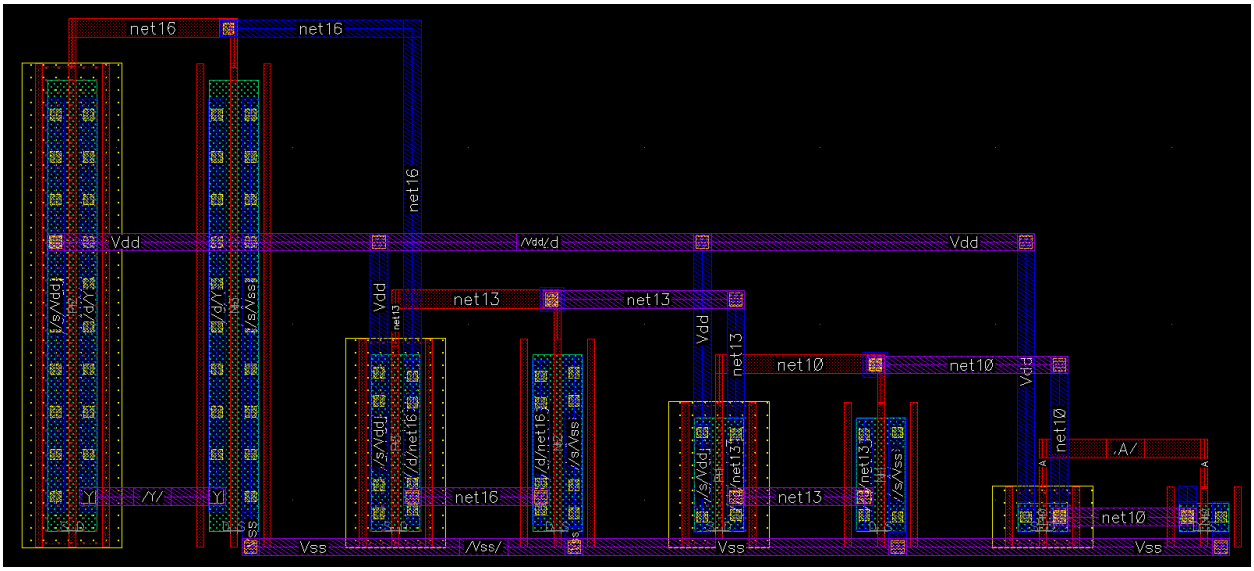


Figure 34 *– Layout of a buffer which drives an input to the column decoder*

The ‘row’ buffers were similar enough in layout that it wasn’t deemed necessary by the author to show an image of its layout.

4.9 12-Bit DAC

Finally, the pad layout and interconnects of the I/O can be seen in Figure 35. Note that the blue text box labeled ‘DAC’ contains the full DAC circuits discussed above, which are not show directly due to issues with rendering at this level of magnification.

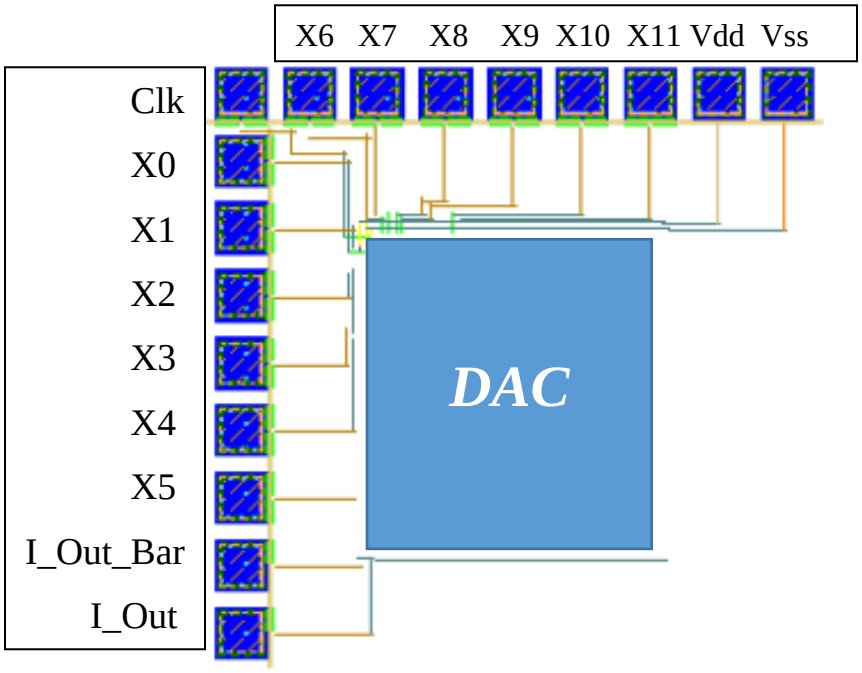


Figure 35 *– DAC layout showing location of bondpads and their respective purpose*

4.10 8-Bit DAC

An 8-bit DAC is also included on the test-chip, and an image of the DAC with bondpads and routing can be seen in Figure 36.

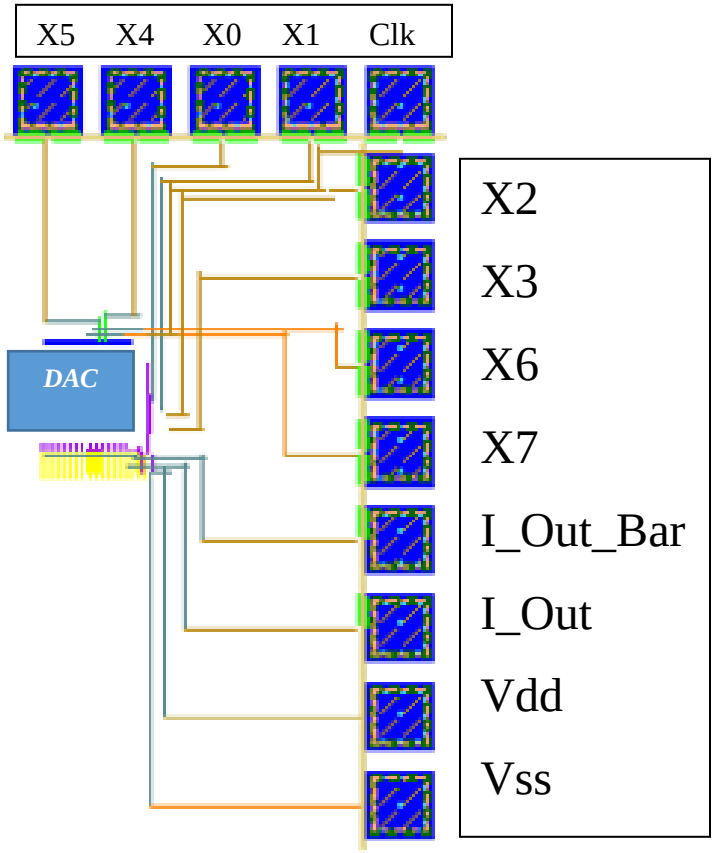


Figure 36 *– The 8-bit DAC's I/O*

4.11 Full Layout of the Test-Chip

This 12-bit DAC, as mentioned previously, is a submodule of a larger test-chip which was sent out for fabrication. For the sake of completeness, the full I/O of the test-chip is shown in Figure 37.

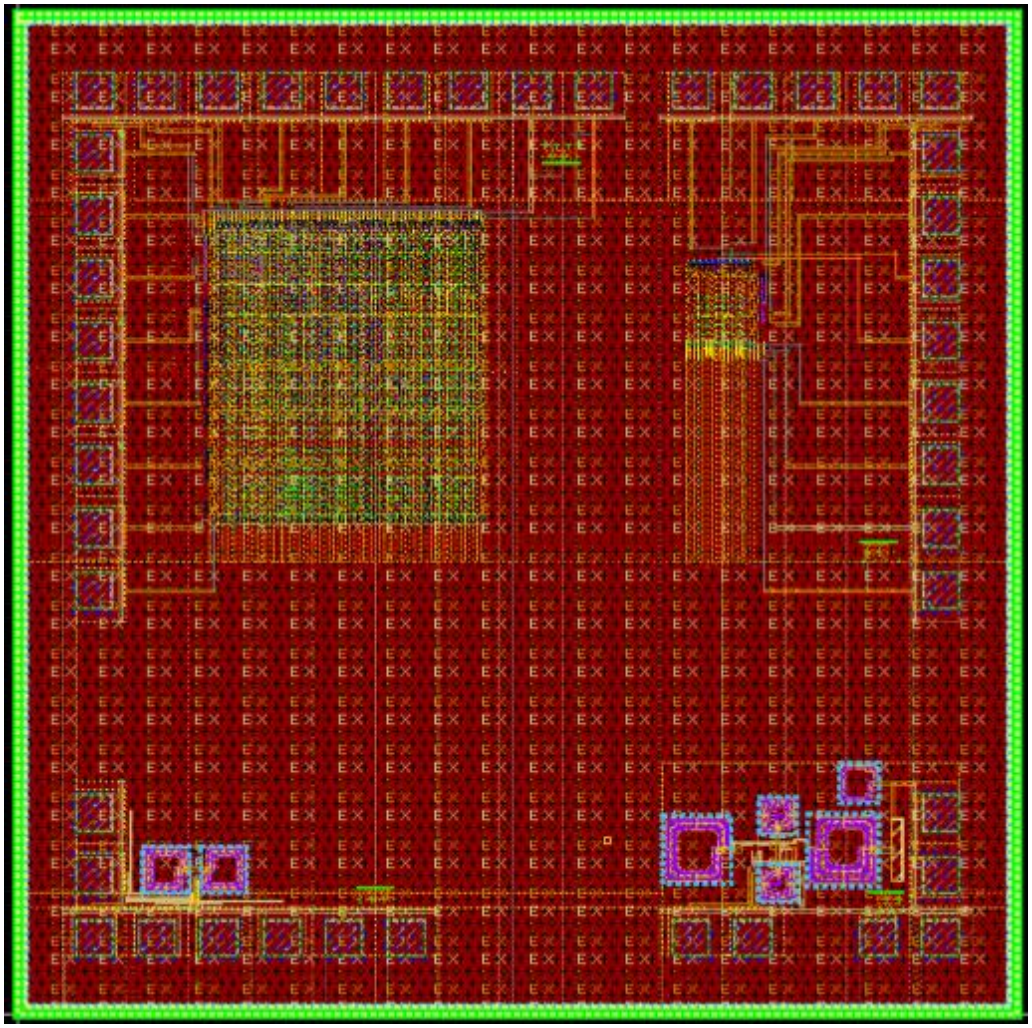


Figure 37 *– The full test-chip that was sent out for fabrication in May of 2020*

Chapter 5 - Simulation Results & Verification

This section will examine the performance of the various submodules of the overall design, as well as simulations of the fully functional DAC. Figures of waveforms are provided, as well as figures of testbench schematics.

5.1 2-4 Binary-Thermometer Decoder

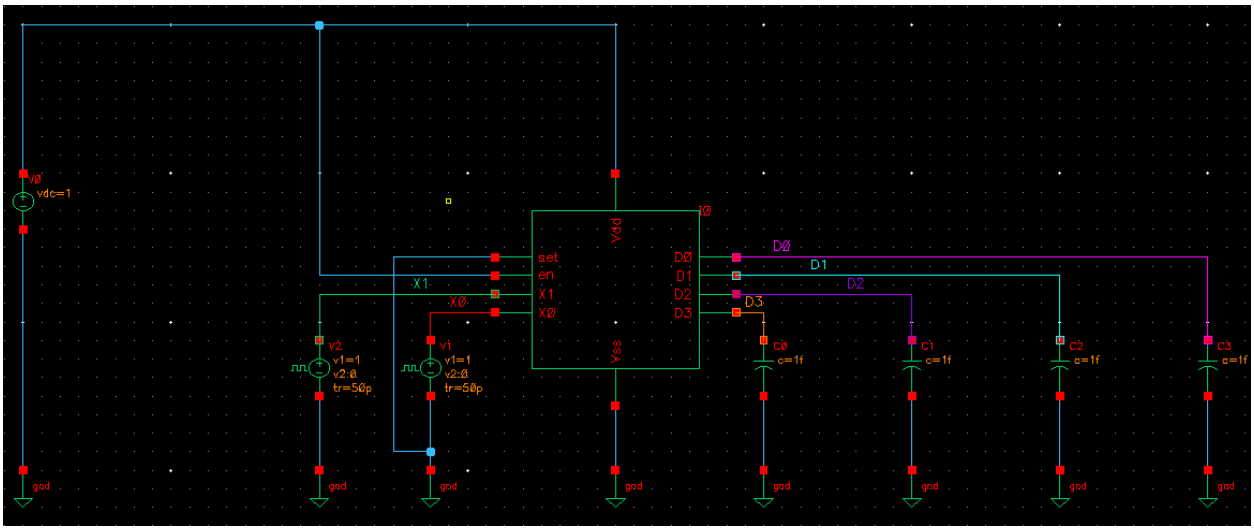


Figure 38 *– The testbench schematic designed to test the 2-4 binary-thermometer decoder*

Table 2 provides a comprehensive explanation of the input stimuli. Table 3 describes the load each output drives. Table 4 provides a legend for the waveforms shown in Figure 39.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Edge Speed
En	High	Enables device	Vdd – 1 V	NA	NA	NA
Set	High	Sets outputs to 1	Vss – 0 V	NA	NA	NA
X0	High	LSB	Vpulse – analogLib	10n	4.95n	50p
X1	High	MSB	Vpulse – analogLib	20n	9.95n	50p
Vdd	NA	DC Power (1 V)	Vdc - analogLib	NA	NA	NA
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA

Table 2 *– Description of the input pins and their stimuli in the testbench schematic in Figure 38*

Pin	Description	Load on Pin	Load Value
D0	LSB	Cap – analogLib	1 f
D1	MSB - 2	Cap - analogLib	1 f
D2	MSB - 1	Cap - analogLib	1 f
D3	MSB	Cap - analogLib	1 f

Table 3 *– Description of the output pins and their loads in the testbench schematic in Figure 38*

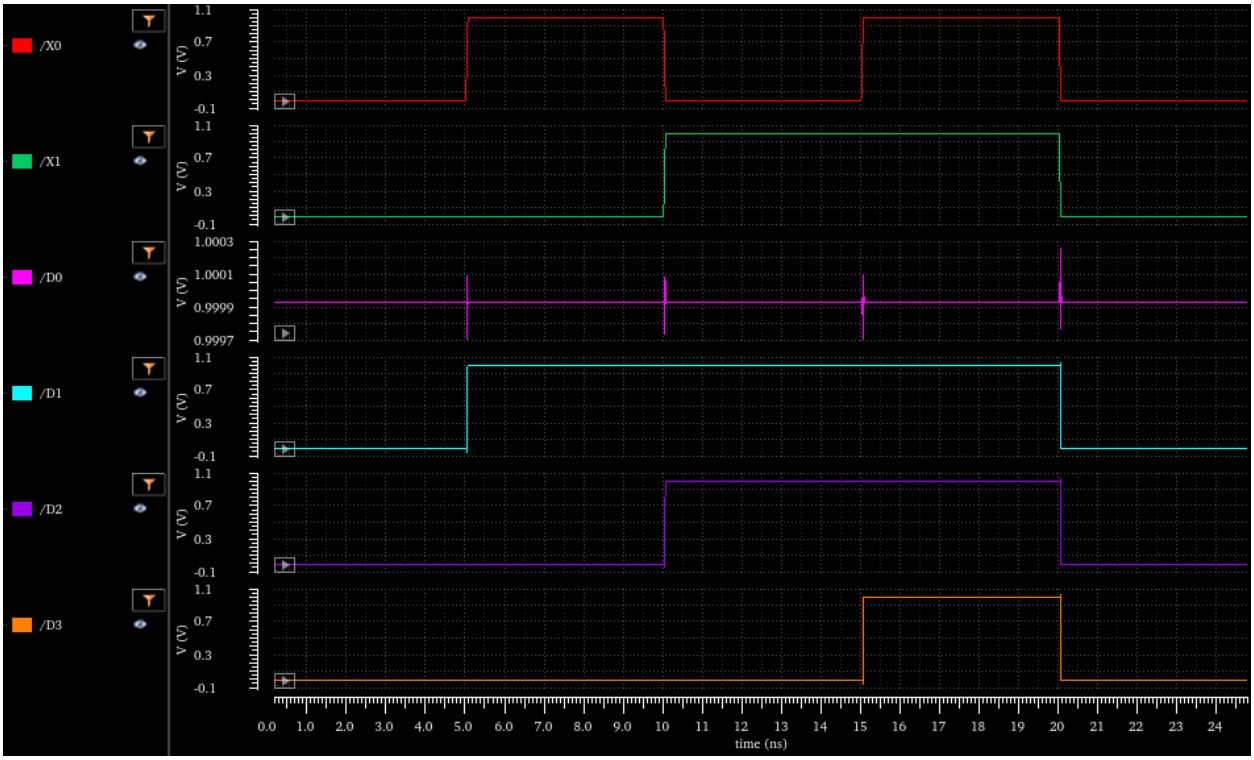


Figure 39 *– Results of the described testbench simulation*

Waveform	Color	Duration of Simulation
X0	Red	25n
X1	Green	
D0	Pink	
D1	Teal	
D2	Purple	
D3	Orange	

Table 4 *– Legend for Figure 39*

5.2 4-16 Binary-Thermometer Decoder

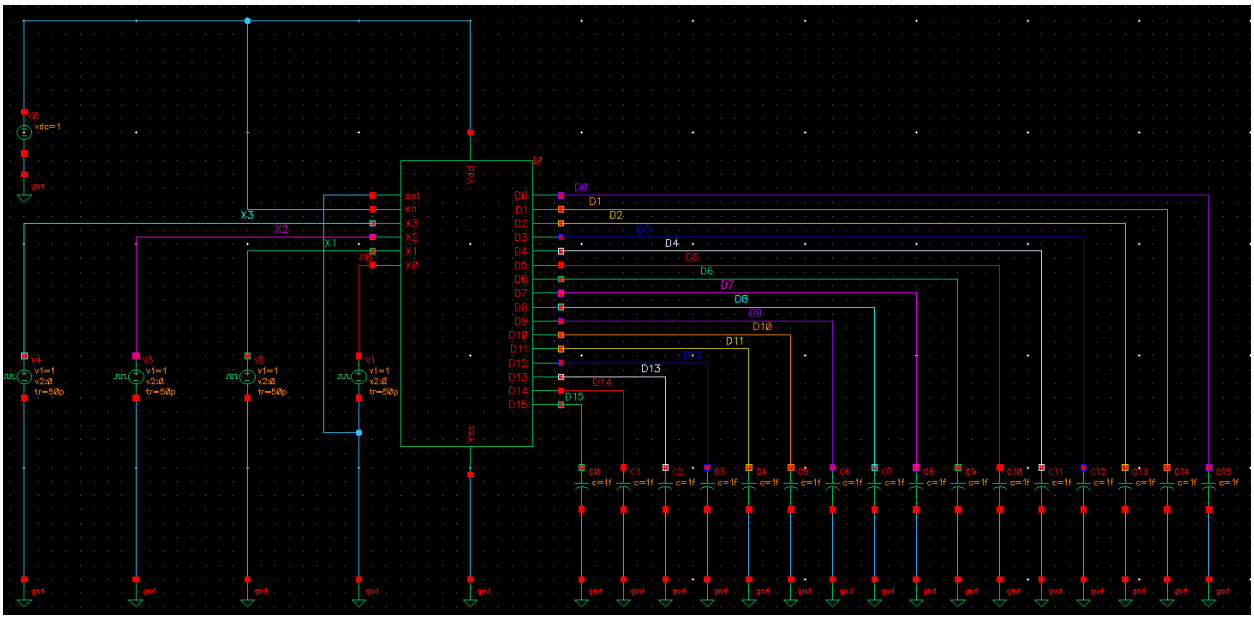


Figure 40 *– The testbench schematic designed to test the 4-16 binary-thermometer decoders*

Table 5 provides a comprehensive explanation of the input stimuli. Table 6 describes the load each output drives. Table 7 provides a legend for the waveforms shown in Figure 41.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Edge Speed
En	High	Enables device	Vdd – 1 V	NA	NA	NA
Set	High	Sets outputs to 1	Vss – 0 V	NA	NA	NA
X0	High	LSB	Vpulse – analogLib	10n	4.95n	50p
X1	High	MSB - 2	Vpulse – analogLib	20n	9.95n	50p
X2	High	MSB - 1	Vpulse – analogLib	40n	19.95n	50p
X3	High	MSB	Vpulse – analogLib	80n	39.95n	50p
Vdd	NA	DC Power (1 V)	Vdc - analogLib	NA	NA	NA
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA

Table 5 *- Description of the input pins and their stimuli in the testbench schematic in Figure 40*

Pin	Description	Load on Pin	Load Value
D0	LSB	Cap – analogLib	1 f
D1	MSB – 14	Cap - analogLib	1 f
D2	MSB – 13	Cap - analogLib	1 f
D3	MSB – 12	Cap - analogLib	1 f
D4	MSB – 11	Cap – analogLib	1 f
D5	MSB – 10	Cap – analogLib	1 f
D6	MSB – 9	Cap – analogLib	1 f
D7	MSB – 8	Cap – analogLib	1 f
D8	MSB – 7	Cap – analogLib	1 f
D9	MSB – 6	Cap – analogLib	1 f
D10	MSB – 5	Cap – analogLib	1 f
D11	MSB – 4	Cap – analogLib	1 f
D12	MSB – 3	Cap – analogLib	1 f
D13	MSB – 2	Cap – analogLib	1 f
D14	MSB – 1	Cap – analogLib	1
D15	MSB	Cap – analogLib	1 f

Table 6 *– Description of the output pins and their loads in the testbench schematic in Figure 40*

Figure 41 *- Results of the described testbench simulation*

Waveform	Color	Duration of Simulation
X0	Red	95n
X1	Green	
X2	Pink	
X3	Teal	
D0	Purple	
D1	Orange	
D2	Yellow	
D3	Blue	
D4	White	
D5	Red	
D6	Green	
D7	Pink	
D8	Blue	
D9	Purple	
D10	Orange	
D11	Yellow	
D12	Blue	
D13	White	
D14	Red	
D15	Green	

Table 7 *– Legend for Figure 41*

5.3 6-64 Binary-Thermometer Decoder

Please note that not all 64 outputs are shown for the sake of readability in Figure 42 – several outputs were selected which confirm functionality of the circuit. Also, no schematic is shown due to the large size of the decoder's testbench – the image shows poor resolution and is not helpful in understanding the test conditions – refer to the previous testbench images, they were mimicked for the 6-64 decoder.

Table 8 describes the inputs and their stimuli, while Table 9 describes the outputs plotted in Figure 42 and their respective loads. Table 10 provides a legend for the waveforms plotted in Figure 42.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Edge Speed
X0	High	LSB	Vpulse – analogLib	2n	0.95n	50p
X1	High	MSB - 4	Vpulse – analogLib	4n	1.95n	50p
X2	High	MSB - 3	Vpulse – analogLib	8n	3.95n	50p
X3	High	MSB - 2	Vpulse – analogLib	16n	7.95n	50p
X4	High	MSB - 1	Vpulse – analogLib	32n	15.95n	50p
X5	High	MSB	Vpulse – analogLib	64n	31.95n	50p
Vdd	NA	DC Power (1 V)	Vdc - analogLib	NA	NA	NA
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA

***Table 8** - Description of the input pins and their stimuli in the testbench schematic for the 6-64 decoder*

Pin	Description	Load on Pin	Load Value
D0	LSB	Cap – analogLib	1 f
D1	MSB – 62	Cap - analogLib	1 f
D2	MSB – 61	Cap - analogLib	1 f
D3	MSB – 60	Cap - analogLib	1 f
D4	MSB – 59	Cap – analogLib	1 f
D7	MSB – 56	Cap – analogLib	1 f
D15	MSB – 48	Cap – analogLib	1 f
D31	MSB – 32	Cap – analogLib	1 f
D63	MSB	Cap – analogLib	1 f

Table 9 *- Description of the output pins and their loads in the testbench schematic*

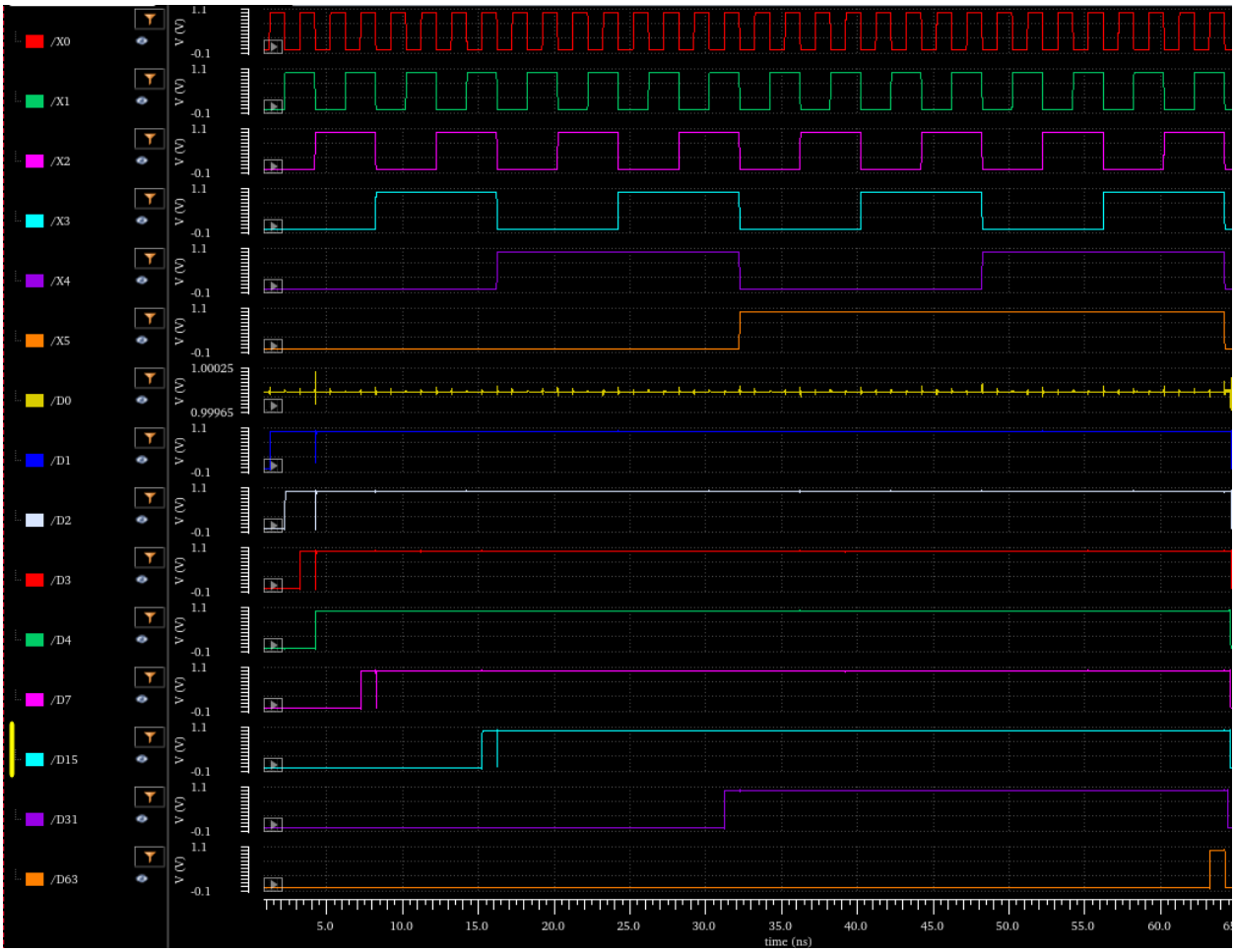


Figure 42 *– Result of the described testbench simulation*

Waveform	Color	Duration of Simulation
X0	Red	65n
X1	Green	
X2	Pink	
X3	Teal	
X4	Purple	
X5	Orange	
D0	Yellow	
D1	Blue	
D2	White	
D3	Red	
D4	Green	
D7	Pink	
D15	Teal	
D31	Purple	
D63	Orange	

Table 10 *– Legend for Figure 42*

5.4 D-Flip-Flop

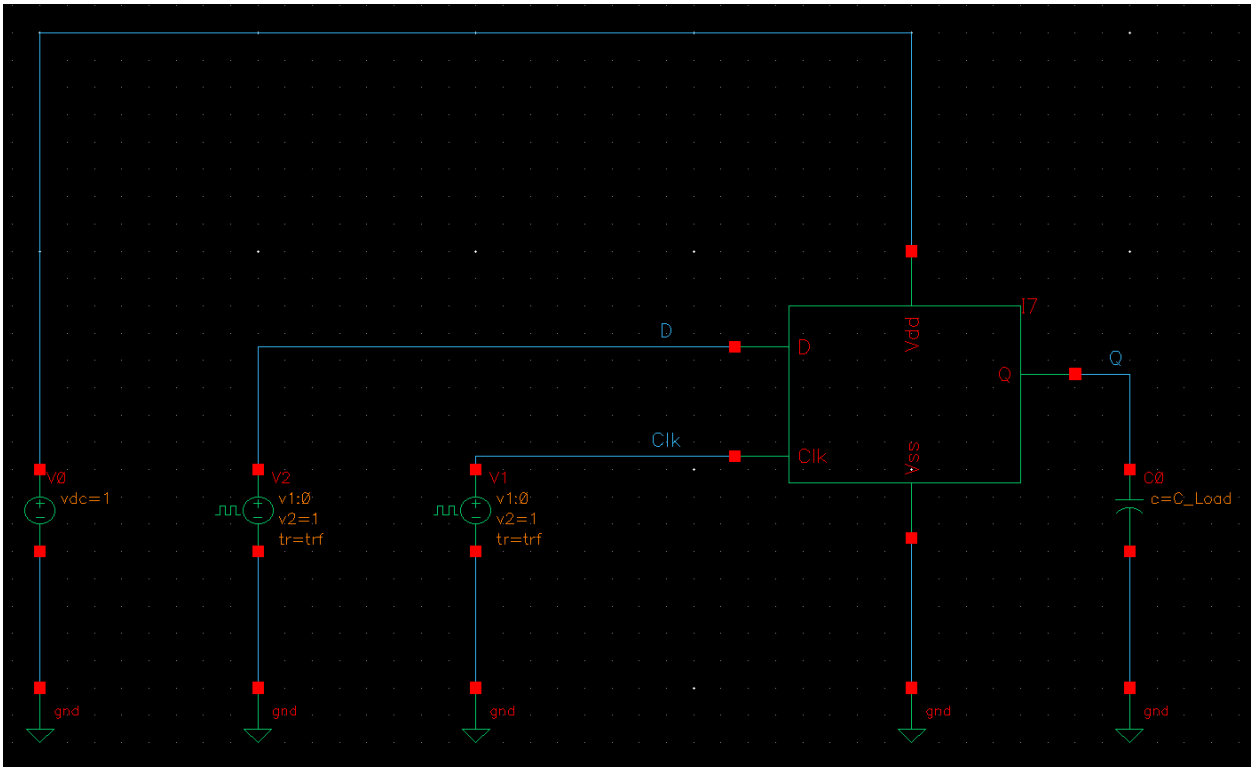


Figure 43 *- The testbench schematic designed to test the D-Flip-Flop*

Table 11 provides definitions for the input pins, as well as data on the stimuli used in the simulation. Table 12 provides a description of the output pin and the load it drives. Figure 44 shows the results of the simulation, and Table 13 provides a legend for Figure 44.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Edge Speed
Clk	High	Triggers DFF on positive edge	Vpulse - analogLib	2n	0.95n	50p
D	High	Data to be clocked	Vpulse - analogLib	12.24n	6.07n	50p
Vdd	NA	DC Power (1 V)	Vdc - analogLib	NA	NA	NA
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA

Table 11 *– Description of the input pins and their stimuli in the testbench schematic in Figure 43*

Pin	Description	Load on Pin	Load Value
Q	Registered output of DFF	Cap – analogLib	500 f

Table 12 *- Description of the output pins and their loads in the testbench schematic in Figure 43*

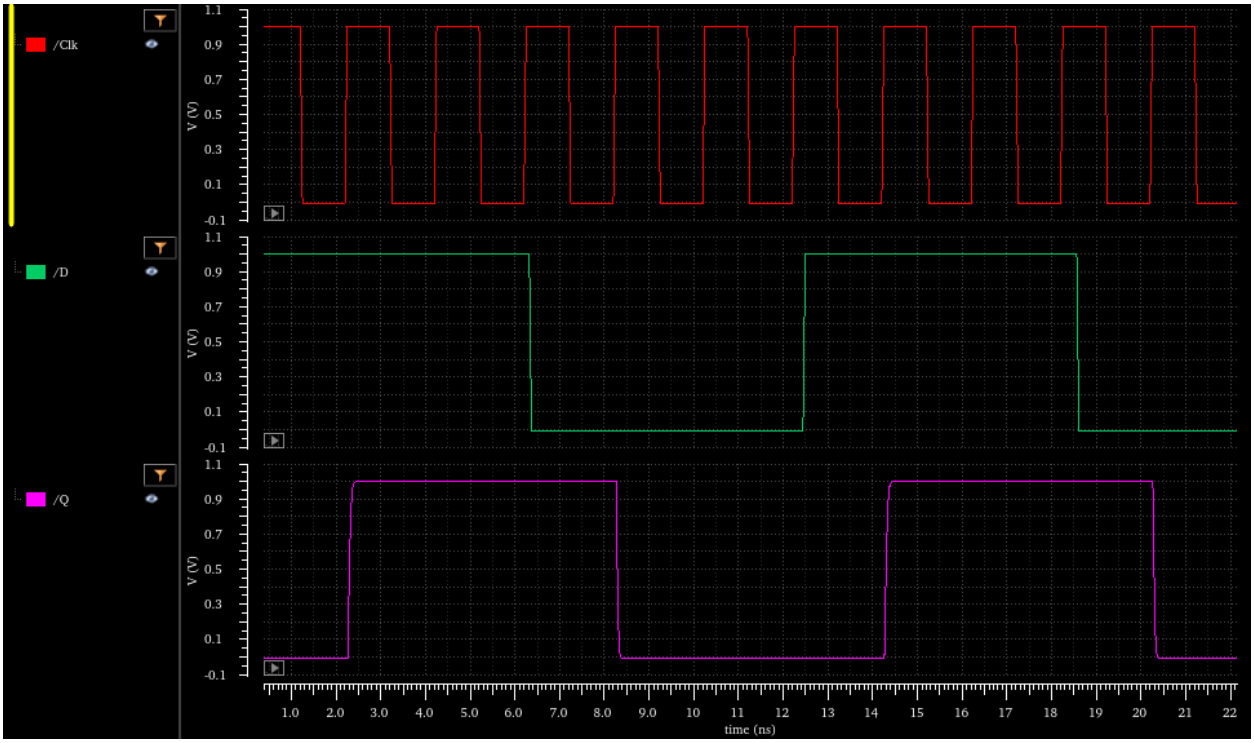


Figure 44 *– Results of the described testbench simulation*

Waveform	Color	Duration of Simulation
Clk	Red	22n
D	Green	
Q	Purple	

Table 13 *– Legend for Figure 44*

5.5 Synchronous Decoder (Outputs of 6-64 Decoder Clocked w/DFFs)

As was the case in 5.3, the testbench schematic is far too large for an image to be included. It consisted of a 6-64 decoder and 64 DFF's, one at each output of the decoder, to register the data. Refer to the table below for the stimuli to each input.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Edge Speed
Clk	High	Clock for DFF's	Vpulse – analogLib	2n	0.97n	30p
X0	High	LSB	Vpulse – analogLib	6.8n	3.37n	30p
X1	High	MSB - 4	Vpulse – analogLib	13.6n	6.77n	30p
X2	High	MSB - 3	Vpulse – analogLib	27.2n	13.57n	30p
X3	High	MSB - 2	Vpulse – analogLib	54.5n	27.17n	30p
X4	High	MSB - 1	Vpulse – analogLib	108.8n	54.47n	30p
X5	High	MSB	Vpulse – analogLib	217.6n	108.8n	30p
Vdd	NA	DC Power (1 V)	Vdc - analogLib	NA	NA	NA
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA

***Table 14** - Description of the input pins and their stimuli in the testbench schematic for the registered 6-64 decoder*

Pin	Description	Load on Pin	Load Value
D0	LSB	Cap – analogLib	1 f
D1	MSB – 62	Cap - analogLib	1 f
D2	MSB – 61	Cap - analogLib	1 f
D3	MSB – 60	Cap - analogLib	1 f
D4	MSB – 59	Cap – analogLib	1 f
D7	MSB – 56	Cap – analogLib	1 f
D15	MSB – 48	Cap – analogLib	1 f
D31	MSB – 32	Cap – analogLib	1 f

***Table 15** - Description of the output pins and their loads from the testbench schematic*

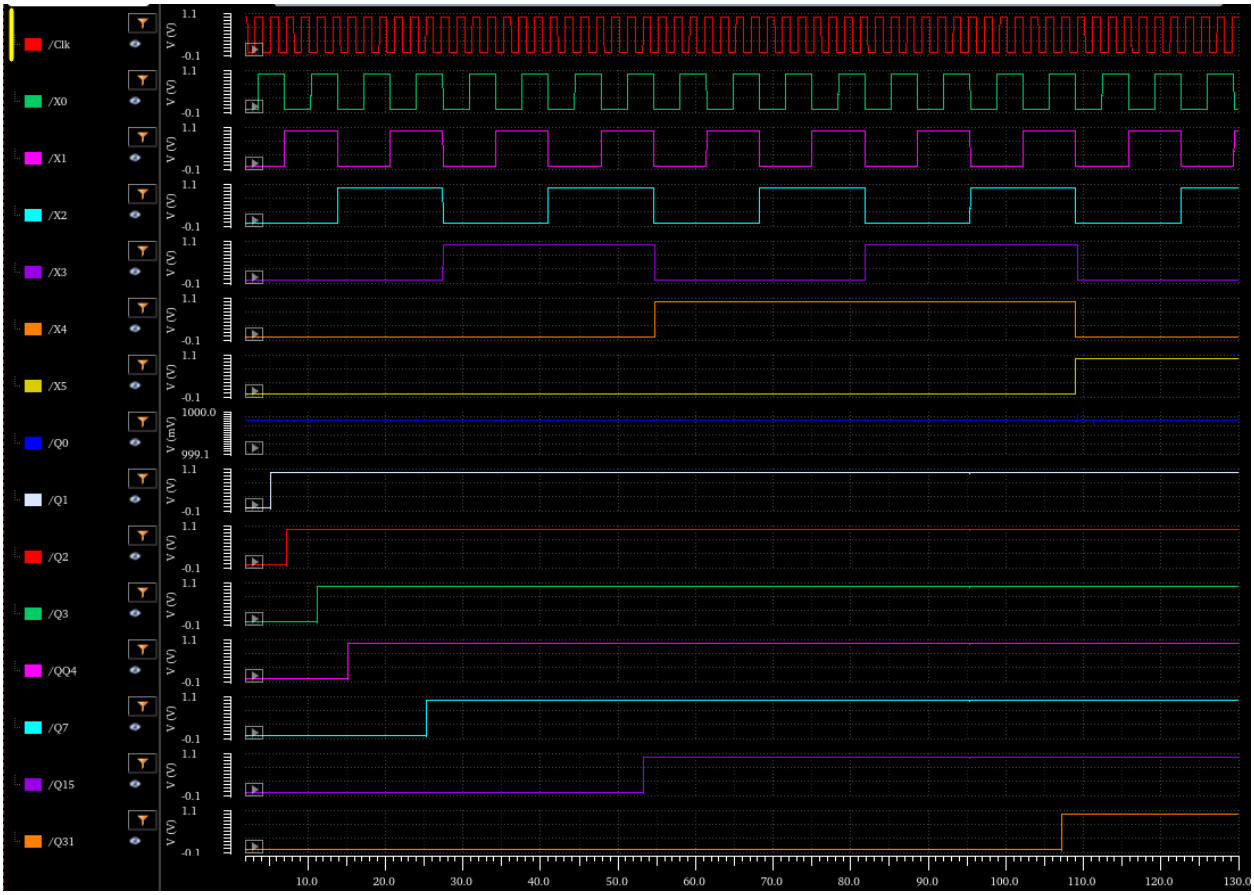


Figure 45 *– Results of the described testbench simulation*

Waveform	Color	Duration of Simulation
Clk	Red	130n
X0	Green	
X1	Pink	
X2	Teal	
X3	Purple	
X4	Orange	
X5	Yellow	
D0	Blue	
D1	White	
D2	Red	
D3	Green	
D4	Pink	
D7	Teal	
D15	Purple	
D31	Orange	

Table 16 *– Legend for Figure 45*

Note that the ‘glitches’ present in the decoder simulations from previous pages are not present when the outputs are registered through a bank of flip-flops – yet another advantage of registering data to the DAC.

5.6 Current Cell

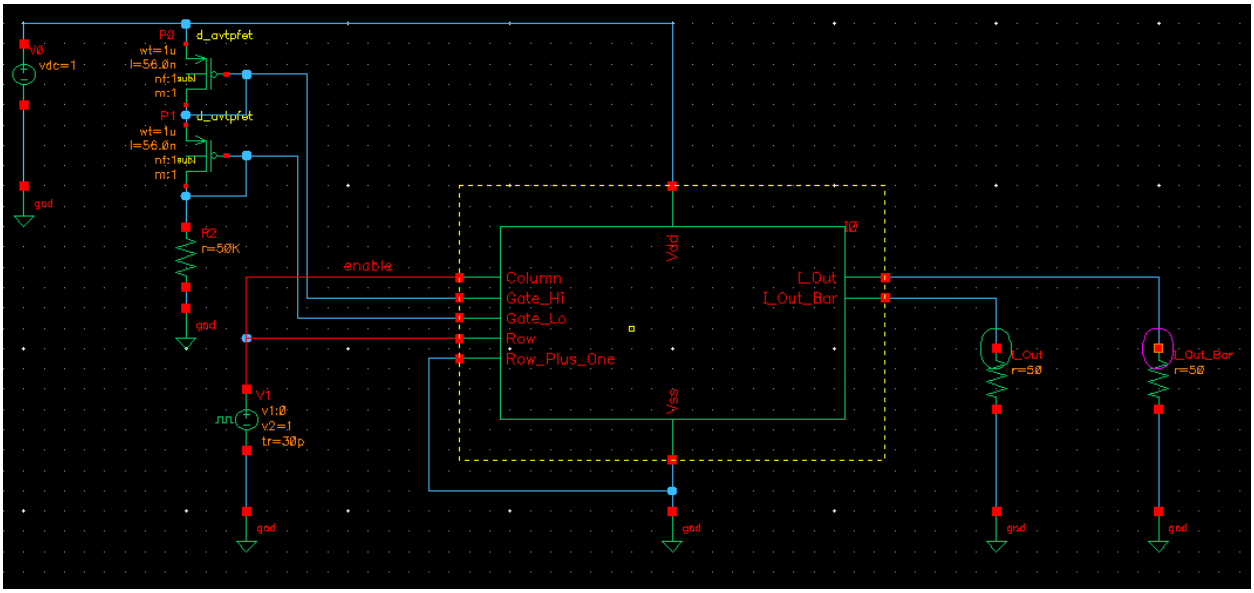


Figure 46 – Testbench schematic created to verify the functionality of the current cell

Figure 43 shows the testbench schematic created to verify the functionality of the current cell. Table 17 provides data regarding the input pins and their stimuli, while Table 18 provides data on the output pins and their respective loads. Figure 47 shows the results of the testbench schematic sim, and Table 19 provides a legend for the waveforms shown in Figure 47.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Edge Speed
Column	High	Digital input	Vpulse – analogLib	2n	0.97n	30p
Row	High	Digital input				
Row_Plus_One	High	Digital input	Gnd - analogLib	NA	NA	NA
Gate_Hi	NA	Bias for cascode	Current Reference	NA	NA	NA
Gate_Lo	NA	Bias for cascode	Current Reference	NA	NA	NA
Vdd	NA	DC Power (1 V)	Vdc - analogLib	NA	NA	NA
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA

***Table 17** - Description of the input pins and their stimuli in the testbench schematic for the current cell*

Pin	Description	Load on Pin	Load Value
I_Out	Complementary current output	Res – analogLib	50 Ohms
I_Out_Bar	Complementary current output	Res - analogLib	50 Ohms

Table 18 *- Description of the output pins and their loads from the testbench schematic*

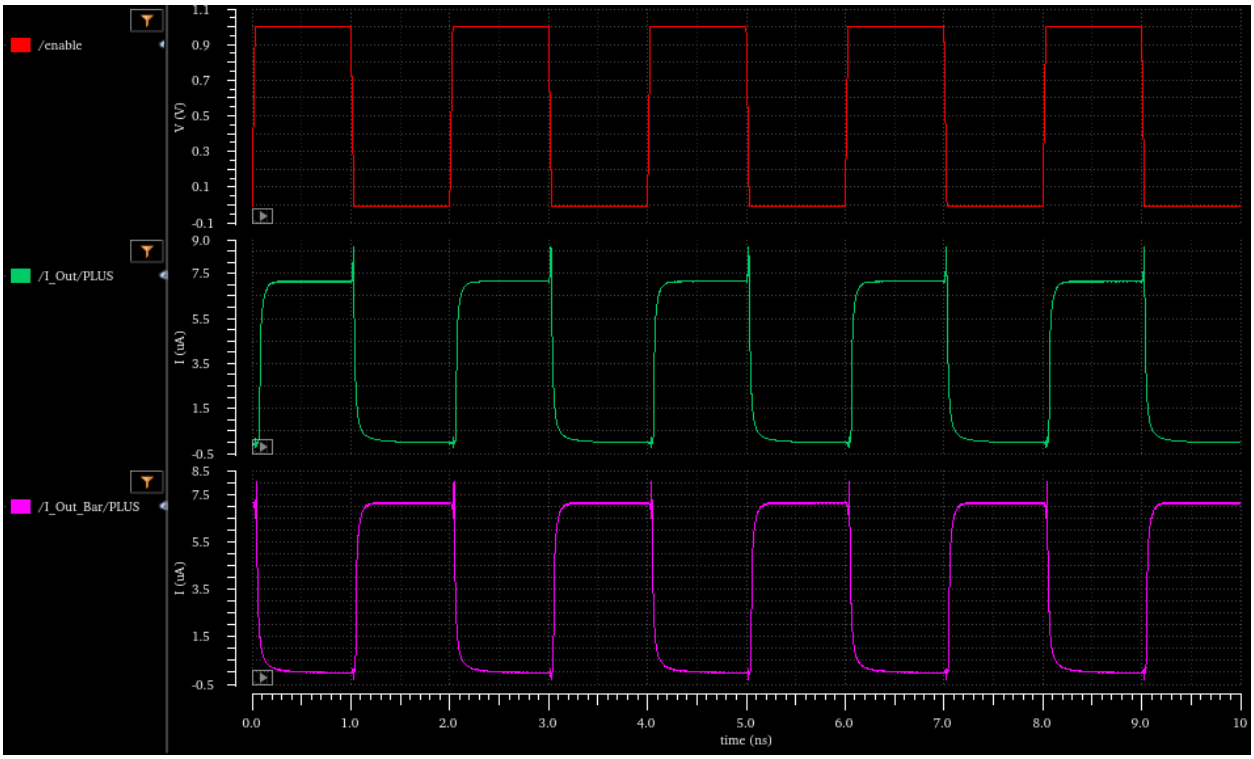


Figure 47 – The results of the testbench schematic

Waveform	Color	Duration of Simulation
enable	Red	10n
I_Out	Green	
I_Out_Bar	Pink	

Table 19 *– A legend for the waveforms shown in Figure 47*

5.7 4-Bit DAC: Linear Ramp of Output Current (Pre-Layout)

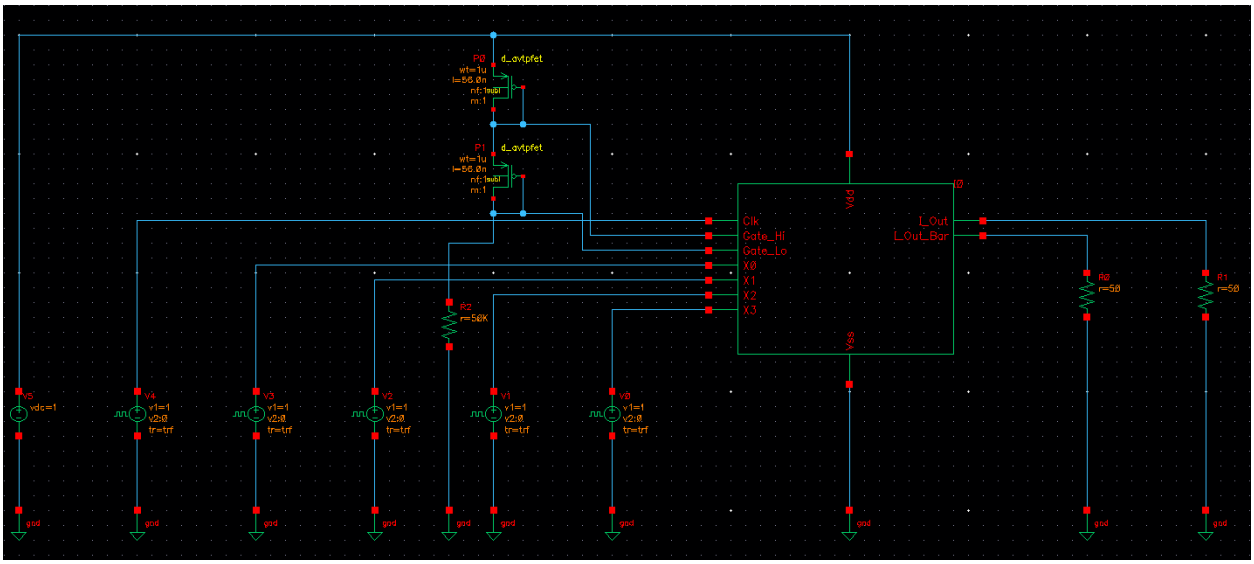


Figure 48 *– The testbench schematic used to verify the 4-bit DAC*

Figure 48 above shows the testbench schematic for the 4-bit DAC. Table 20 gives data pertaining to the inputs and their stimuli, while Table 21 contains data for the outputs and their respective loads. Figure 49 shows the waveforms of interest in the testbench simulation, while Table 22 provides a legend for the Figure 49. Finally, Figure 50 shows the desired output current alone, such that the linear ramp is more pronounced in the image.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Edge Speed
Clk	High	Trigger DFF's	Vpulse - analogLib	2n	1n	30p
X0	High	LSB	Vpulse – analogLib	6.68n	3.34n	30p
X1	High	MSB – 2	Vpulse – analogLib	13.36n	6.68n	30p
X2	High	MSB – 1	Vpulse – analogLib	26.72n	13.36n	30p
X3	High	MSB	Vpulse - analogLib	53.44n	26.72n	30p
Gate_Hi	NA	Bias for cascode	Current Reference	NA	NA	NA
Gate_Lo	NA	Bias for cascode	Current Reference	NA	NA	NA
Vdd	NA	DC Power (1 V)	Vdc - analogLib	NA	NA	NA
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA

***Table 20** - Description of the input pins and their stimuli in the testbench schematic for the current cell*

Pin	Description	Load on Pin	Load Value
I_Out	Complementary current output	Res – analogLib	50 Ohms
I_Out_Bar	Complementary current output	Res - analogLib	50 Ohms

***Table 21** - Description of the output pins and their loads from the testbench schematic*

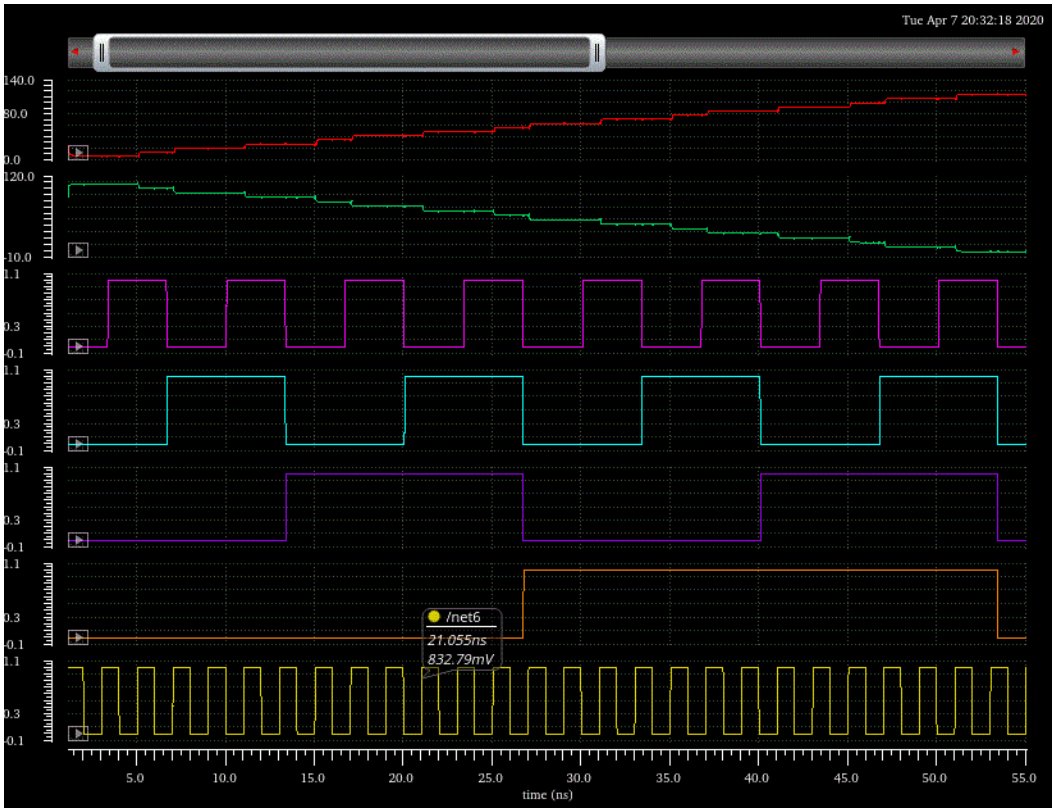


Figure 49 *– Results of the 4-bit DAC linear ramp simulation*

Waveform	Color	Duration of Simulation
I_Out	Red	57n
I_Out_Bar	Green	
X0	Pink	
X1	Teal	
X2	Purple	
X3	Orange	
Clk	Yellow	

Table 22 *– A legend for the waveforms shown in Figure 49*

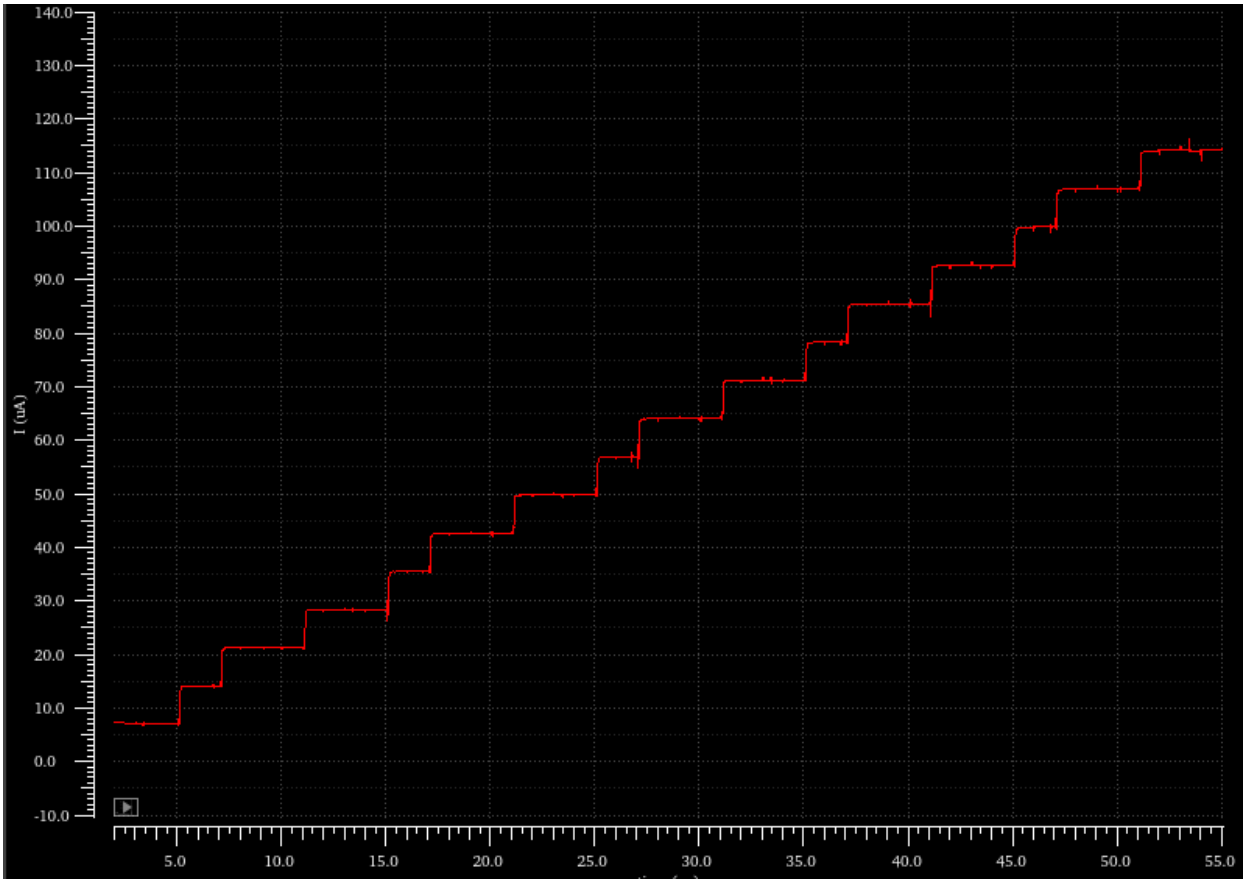


Figure 50 *– I_Out, the current of interest in the 4-bit DAC sim*

5.8 8-bit DAC: Exponential Ramp of Output Current (Pre-Layout)

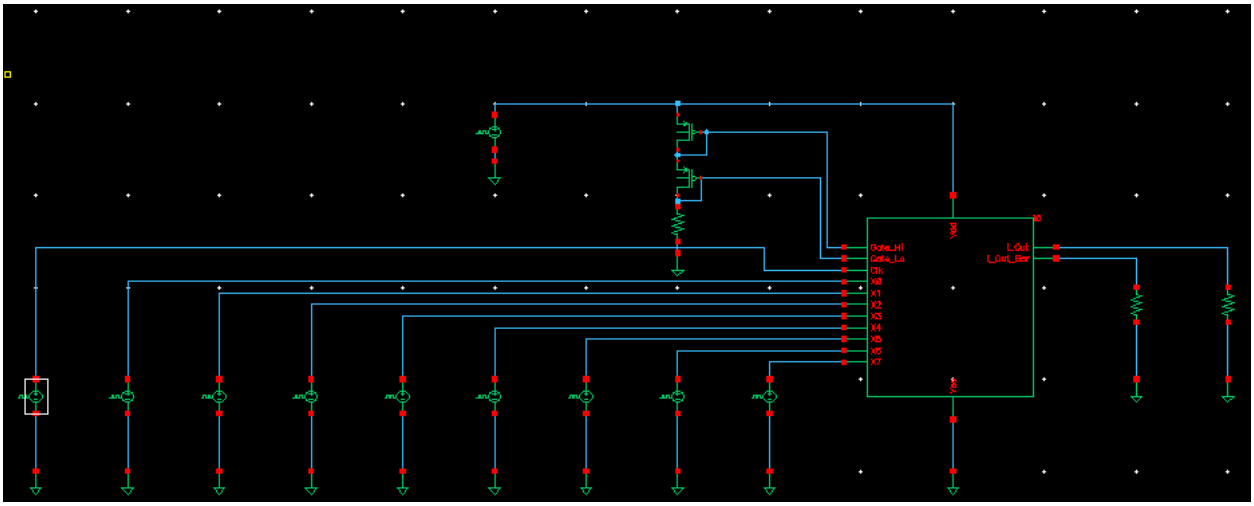


Figure 51 *– The testbench schematic responsible for an exponential ramp current waveform*

Figure 48 shows the testbench which was used for the simulation described in 5.8. Table 23 describes the input pins and their stimuli, while Table 24 describes the output pins and their respective loads. Figure 52 shows the exponentially ramped output current of the 8-bit DAC. The exponential ramp was created by switching one bit at a time from low to high and then holding it high for the rest of the transient simulation.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Delay	Edge Speed
Clk	High	Trigger DFF's	Vpulse - analogLib	2n	1n	25n	30p
X0	High	LSB	Vpulse – analogLib	500m	0	25n	30p
X1	High	MSB – 6	Vpulse – analogLib	500m	0	27.5n	30p
X2	High	MSB – 5	Vpulse – analogLib	500m	0	30n	30p
X3	High	MSB – 4	Vpulse - analogLib	500m	0	32.5n	30p
X4	High	MSB – 3	Vpulse – analogLib	500m	0	35n	30p
X5	High	MSB – 2	Vpulse – analogLib	500m	0	37.5n	30p
X6	High	MSB – 1	Vpulse – analogLib	500m	0	40n	30p
X7	High	MSB	Vpusle – analogLib	500m	0	42.5n	30p
Gate_Hi	NA	Bias for cascode	Current Reference	NA	NA	NA	NA
Gate_Lo	NA	Bias for cascode	Current Reference	NA	NA	NA	NA
Vdd	NA	DC Power (1 V)	Vpulse - analogLib	500m	0	20p	1n
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA	NA

***Table 23** - Description of the input pins and their stimuli in the testbench schematic for 5.8*

Pin	Description	Load on Pin	Load Value
I_Out	Complementary current output	Res – analogLib	50 Ohms
I_Out_Bar	Complementary current output	Res - analogLib	50 Ohms

***Table 24** - Description of the output pins and their loads from the testbench schematic*

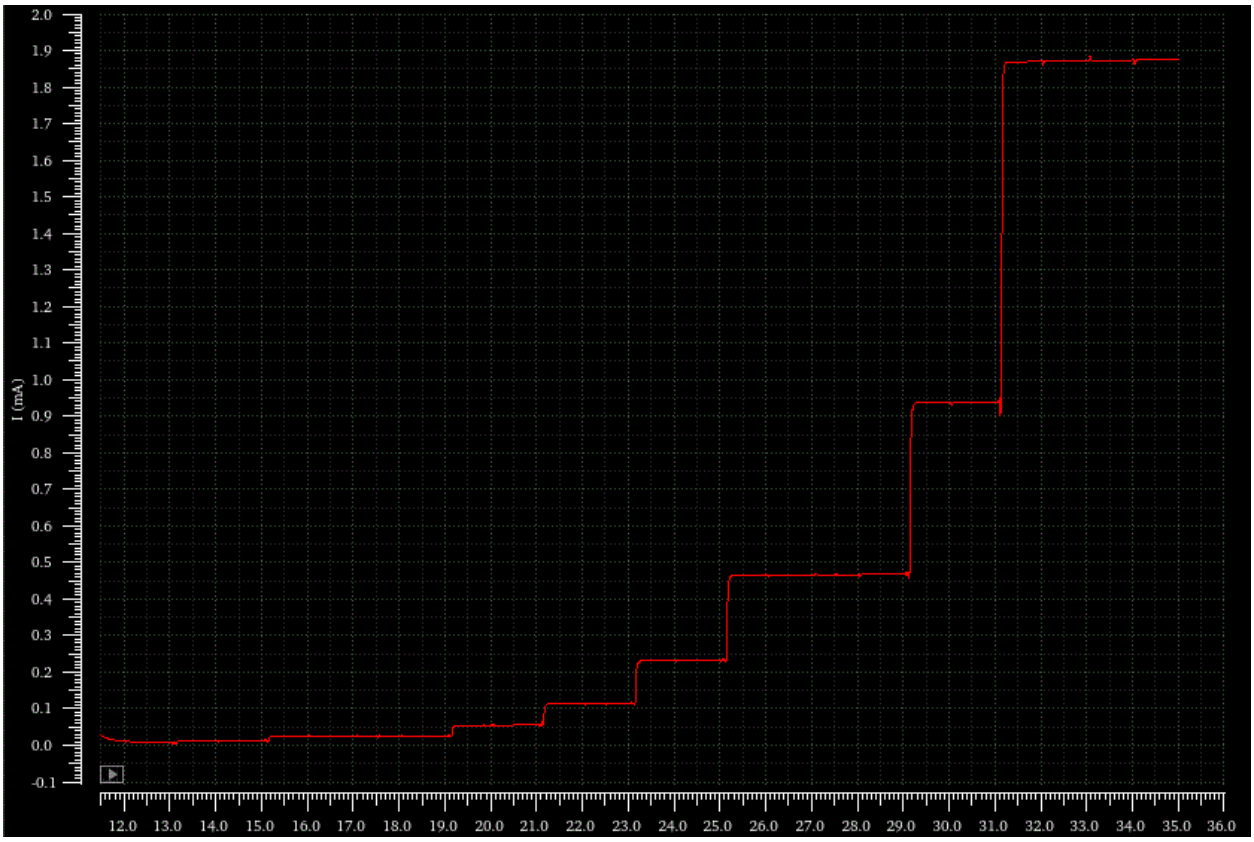


Figure 52 – The exponentially ramped output current of the 8-bit DAC

5.9 8-bit DAC: Linear Ramp of Output Current (Pre-Layout)

Stimuli and the testbench schematic were not documented for this simulation – the green waveform shows the output current of the DAC. Note that ‘glitches’ in the ramp – see 5.10 for a further discussion on this subject.

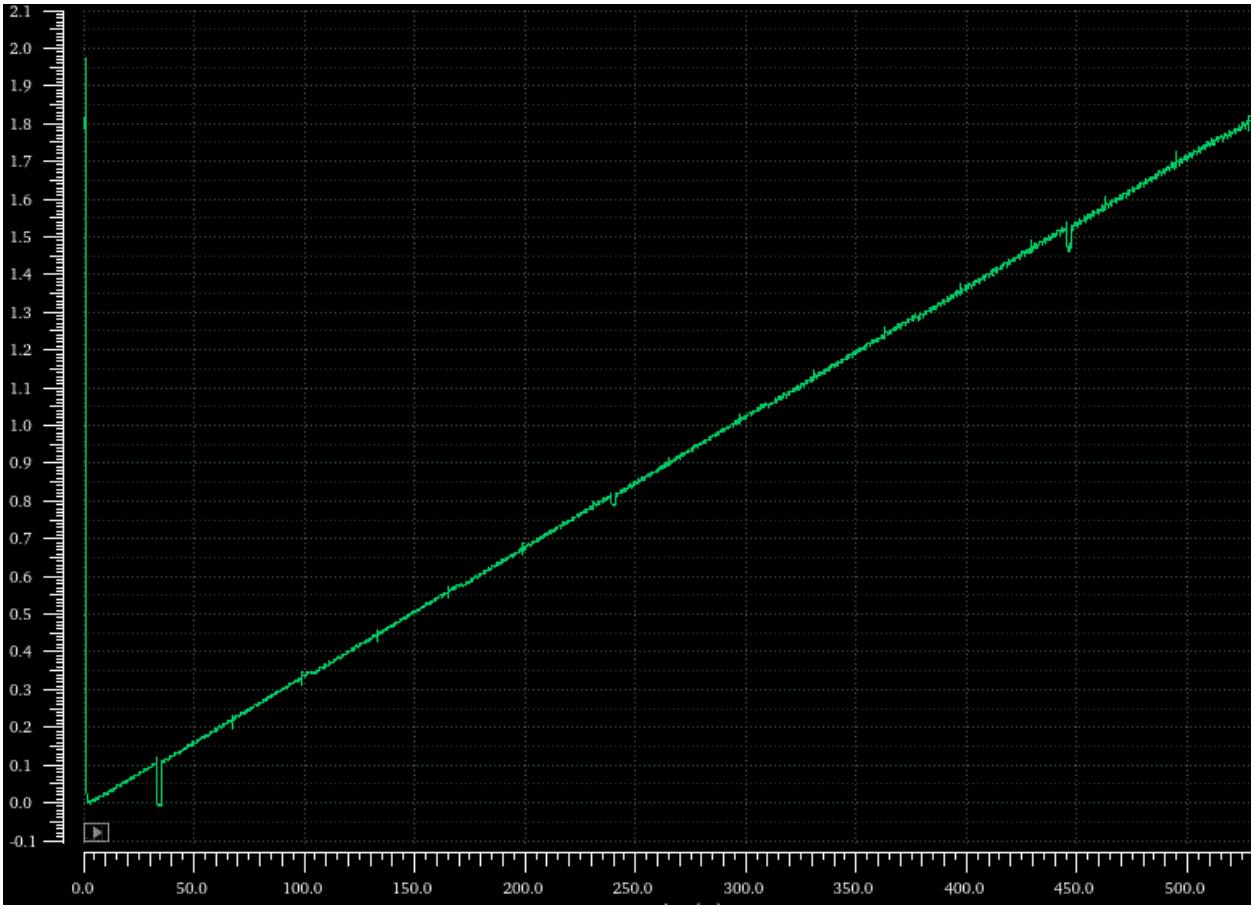


Figure 53 *– Linear ramp of the output current of an 8-bit DAC*

5.10 8-bit DAC Linear Ramp of Output Current w/Large Transient Spikes (Post Layout)

This simulation showed interesting results, hence its inclusion in this chapter. As can be seen in sections 5.7 – 5.9, occurrences of set-up time violations occur in the simulations, causing one-clock cycle delays in bit transitions, or more pronounced 'glitches' which distort the 'ramp' of the output current. To ensure no set-up time violations occurred, the simulation in this section was configured such that all bit transitions occurred on the negative edge of the 500 MHz master clock. This ensured 1ns of stability from the digital inputs before a positive edge of the clock triggered the DFF's – more than enough to ensure no set-up time violations. The results were not as expected, however. See the waveform in Figure 55 – note the massive transient spikes every 32ns. These occurred on the rising edge of the digital input X3. The cause of these massive spikes is not explicitly known as of the writing of this document, but certainly warrants further investigation (they may be an artifact of the simulator), either through more simulation, or testing of the actual chip upon its fabrication and return to Kansas State. Also noteworthy is that this simulation took 57 hours to complete, limiting opportunity to identify the cause.

It should be noted that the digital input X3 is the MSB of the 4-bit nibble that is sent to the column decoder. It appears that when each individual row is 'full', or all of its cells are engaged, these transient spikes in current occur. It's puzzling that these spikes are not seen in other simulations where data is not changing on the negative clock edge. There is a possibility that some sort of timing violation is occurring specifically on the rising edge of X3. As this is an extracted simulation with a very large number of total nodes, the internal nodes of the simulation were not saved or plotted. The clock waveform coming out of the buffer and fanning out to all D-Flip-Flops may hold some key information as to why these spikes are occurring, but the waveform in question was of course not probed in this simulation.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Delay	Edge Speed
Clk	High	Trigger DFF's	Vpulse - analogLib	2n	0.97n	10n	30p
X0	High	LSB	Vpulse – analogLib	4n	1.97n	21n	30p
X1	High	MSB – 6	Vpulse – analogLib	8n	3.97n	21n	30p
X2	High	MSB – 5	Vpulse – analogLib	16n	7.97n	21n	30p
X3	High	MSB – 4	Vpulse - analogLib	32n	15.97n	21n	30p
X4	High	MSB – 3	Vpulse – analogLib	64n	31.97n	21n	30p
X5	High	MSB – 2	Vpulse – analogLib	128n	63.97n	21n	30p
X6	High	MSB – 1	Vpulse – analogLib	216n	128n	21n	30p
X7	High	MSB	Vpusle – analogLib	532n	256n	21n	30p
Gate_Hi	NA	Bias for cascode	Current Reference	NA	NA	NA	NA
Gate_Lo	NA	Bias for cascode	Current Reference	NA	NA	NA	NA
Vdd	NA	DC Power (1 V)	Vpulse - analogLib	500m	0	20p	1n
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA	NA

***Table 25** - Description of the input pins and their stimuli in the testbench schematic for 5.10*

Pin	Description	Load on Pin	Load Value
I_Out	Complementary current output	Res – analogLib	50 Ohms
I_Out_Bar	Complementary current output	Res - analogLib	50 Ohms

***Table 26** - Description of the output pins and their loads from the testbench schematic*

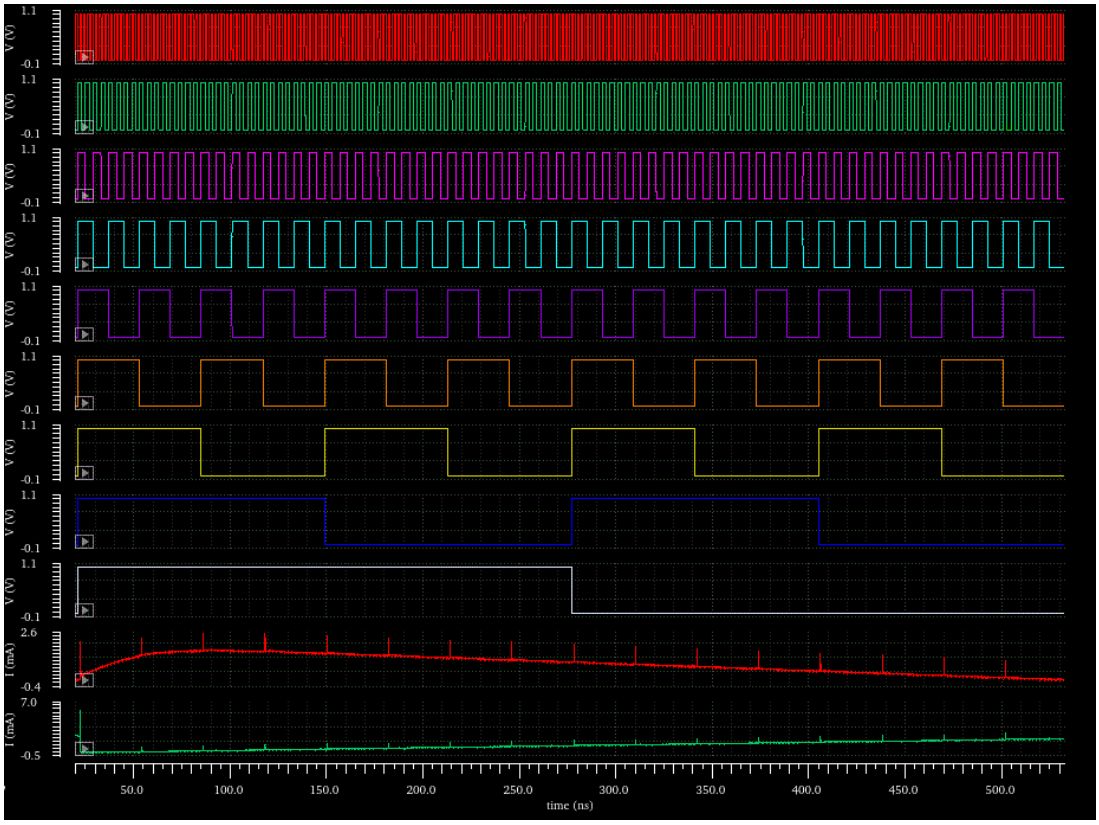


Figure 54 *– The input stimuli and output current waveforms of the 8-bit DAC linear ramp sim*

Waveform	Color	Duration
Clk	Red	530n
X0	Green	
X1	Pink	
X2	Teal	
X3	Purple	
X4	Orange	
X5	Yellow	
X6	Blue	
X7	White	
I_Out_Bar	Red	
I_Out	Green	

Table 27 *– A legend for the waveforms shown in Figure 46*

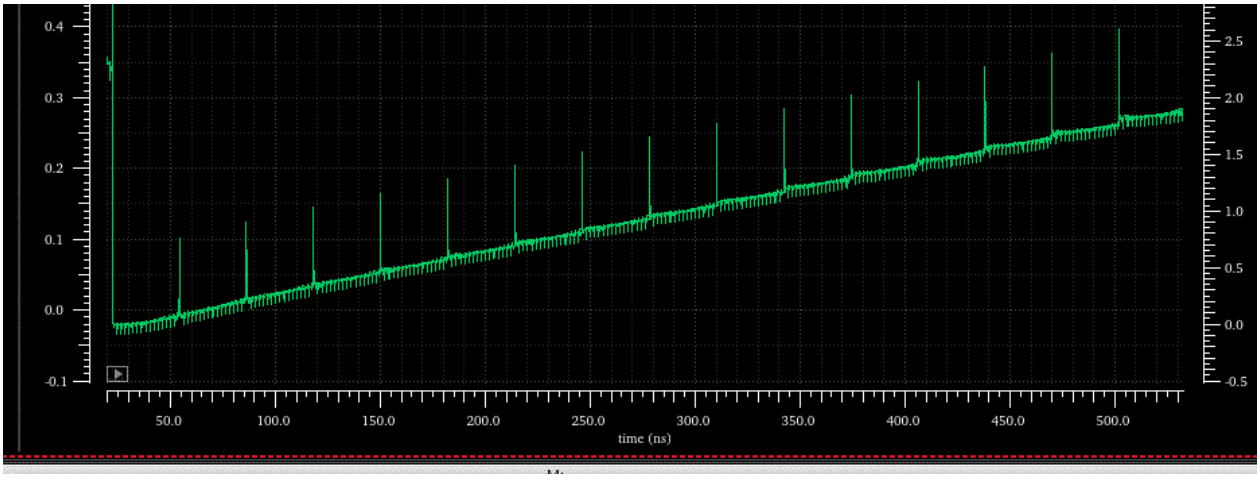


Figure 55 *– The output current of the DAC – note the massive transient spikes every 32ns*

5.11 8-bit DAC Attempted Exponential Ramp of Output Current (Post-Layout)

An attempt was made to generate an exponential ramp in a post-layout transient simulation in the manner of section 5.8, which was a pre-layout simulation. This simulation showed disturbing results, however. It is documented below.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Delay	Edge Speed
Clk	High	Trigger DFF's	Vpulse - analogLib	2n	1n	20n	30p
X0	High	LSB	Vpulse – analogLib	500m	0	30n	30p
X1	High	MSB – 6	Vpulse – analogLib	500m	0	32.5n	30p
X2	High	MSB – 5	Vpulse – analogLib	500m	0	35n	30p
X3	High	MSB – 4	Vpulse - analogLib	500m	0	37.5n	30p
X4	High	MSB – 3	Vpulse – analogLib	500m	0	40n	30p
X5	High	MSB – 2	Vpulse – analogLib	500m	0	42.5n	30p
X6	High	MSB – 1	Vpulse – analogLib	500m	0	45n	30p
X7	High	MSB	Vpusle – analogLib	500m	0	47.5n	30p
Gate_Hi	NA	Bias for cascode	Current Reference	NA	NA	NA	NA
Gate_Lo	NA	Bias for cascode	Current Reference	NA	NA	NA	NA
Vdd	NA	DC Power (1 V)	Vpulse - analogLib	500m	0	20p	1n
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA	NA

***Table 28** - Description of the input pins and their stimuli in the testbench schematic for 5.11*

Pin	Description	Load on Pin	Load Value
I_Out	Complementary current output	Res – analogLib	50 Ohms
I_Out_Bar	Complementary current output	Res - analogLib	50 Ohms

***Table 29** - Description of the output pins and their loads from the testbench schematic*

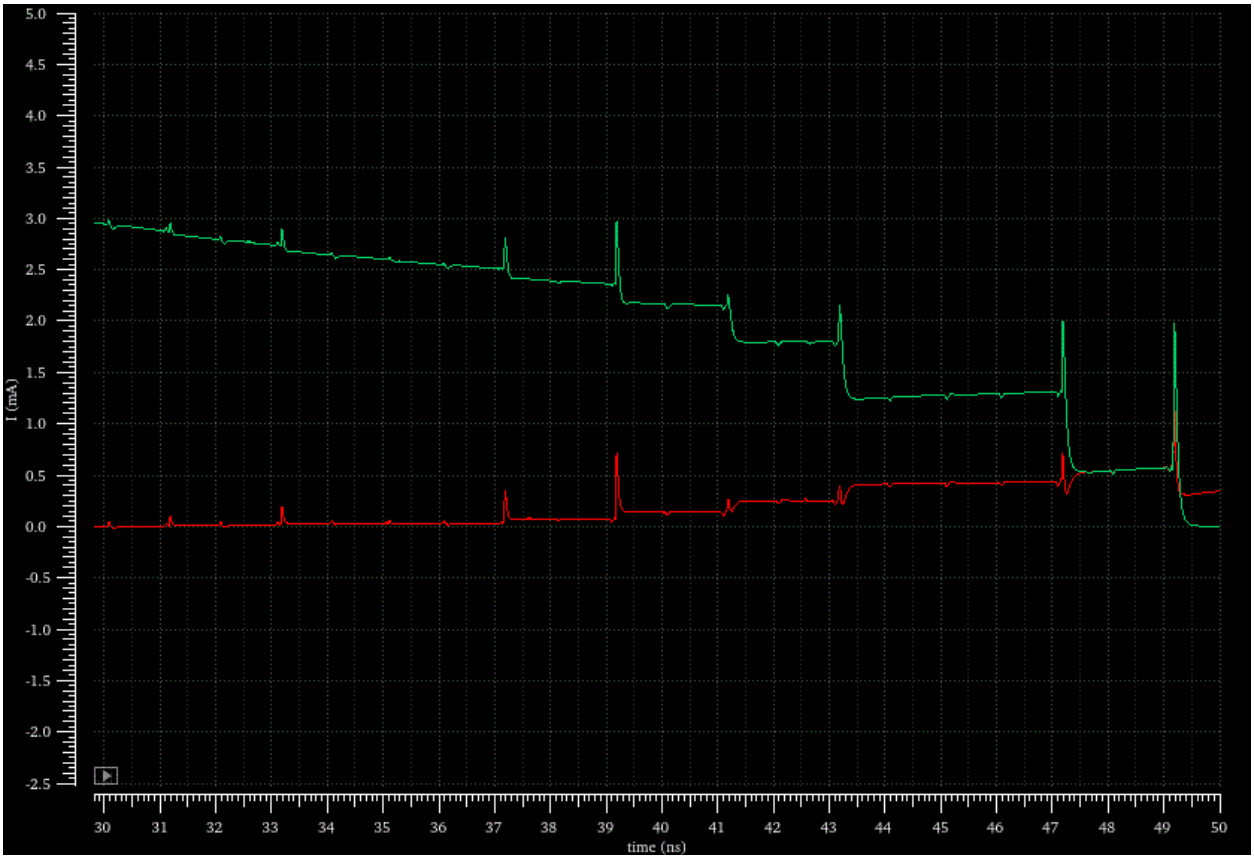


Figure 56 *– Green shows I_Out_Bar while red shows I_Out – clearly something went wrong*

5.12 12-Bit DAC Exponential Ramp (Pre-Layout)

It was noted during earlier exponential ramp simulations of the 12-bit DAC that 'steady-state' current levels (no changing) were not always steady-state, especially when higher order bits were engaged. For this reason, the time between bits switching 'on' was increased significantly.

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Edge Speed
Clk	High	Trigger DFF's	Vpulse - analogLib	2n	1n	30p
X0	High	LSB	Vpulse – analogLib	500m	0	30p
X1	High	MSB – 10	Vpulse – analogLib	500m	0	30p
X2	High	MSB – 9	Vpulse – analogLib	500m	0	30p
X3	High	MSB – 8	Vpulse - analogLib	500m	0	30p
X4	High	MSB – 7	Vpulse – analogLib	500m	0	30p
X5	High	MSB – 6	Vpulse – analogLib	500m	0	30p
X6	High	MSB – 5	Vpulse – analogLib	500m	0	30p
X7	High	MSB – 4	Vpusle – analogLib	500m	0	30p
X8	High	MSB – 3	Vpulse – analogLib	500m	0	30p
X9	High	MSB – 2	Vpulse – analogLib	500m	0	30p
X10	High	MSB – 1	Vpulse – analogLib	500m	0	30p
X11	High	MSB	Vpusle – analogLib	500m	0	30p
Gate_Hi	NA	Bias for cascode	Current Reference	NA	NA	NA
Gate_Lo	NA	Bias for cascode	Current Reference	NA	NA	NA
Vdd	NA	DC Power (1 V)	Vpulse - analogLib	500m	0	1n
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA

Table 30 *– Input pins and stimuli for the 12-bit DAC simulation*

Pin	Description	Load on Pin	Load Value
I_Out	Complementary current output	Res – analogLib	10 Ohms
I_Out_Bar	Complementary current output	Res - analogLib	10 Ohms

Table 31 *- Description of the output pins and their loads from the testbench schematic*

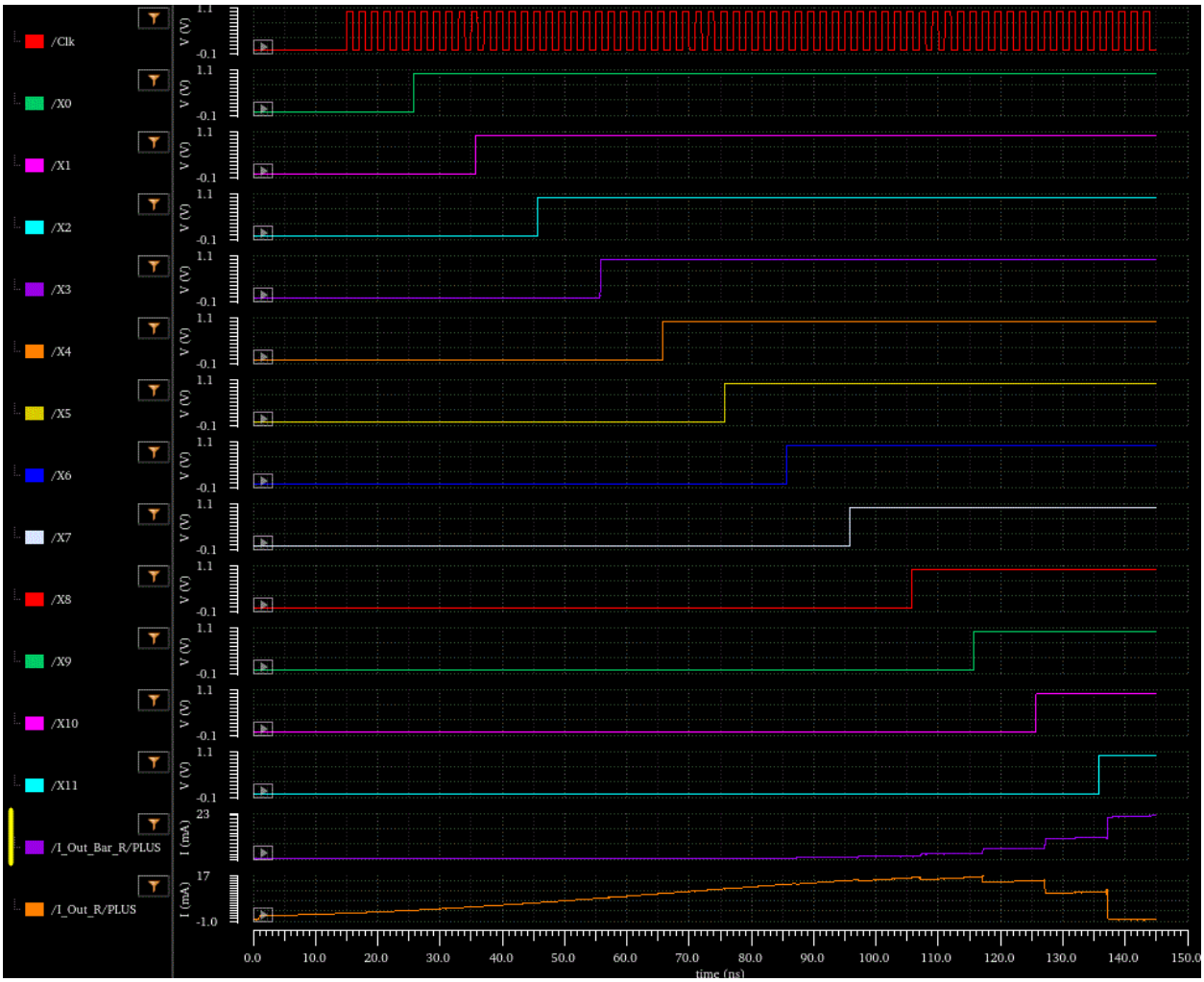


Figure 57 *– All input and output waveforms for the 12-bit DAC sim*

Waveform	Color	Duration
Clk	Red	145n
X0	Green	
X1	Pink	
X2	Teal	
X3	Purple	
X4	Orange	
X5	Yellow	
X6	Blue	
X7	White	
X8	Red	
X9	Green	
X10	Pink	
X11	Teal	
I_Out_Bar	Purple	
I_Out	Orange	

Table 32 *– A legend for the waveforms shown in Figure 46*

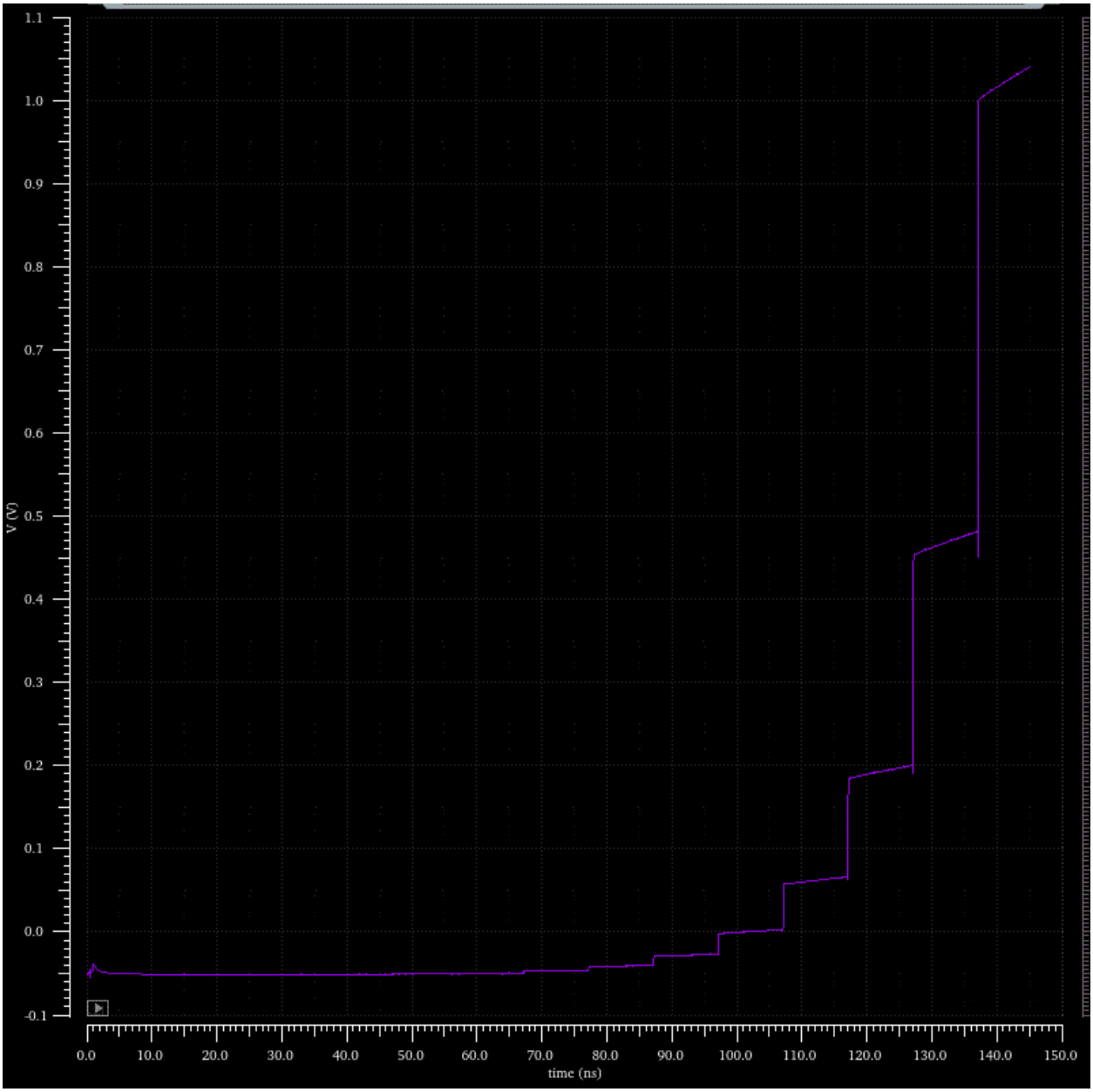


Figure 58 – *The exponentially ramped output current for the 12-bit DAC*

The output current seen in Figure 58 does not reach the expected maximum value of 30mA – it peaks around 22mA. It can also be clearly seen in Figure 58 that the output current does not hold steady when some of the higher order bits are engaged.

The load resistance of the complementary current outputs was changed from 50 Ω to 10 Ω due to the need to drive 30mA with a 1 V power supply voltage.

5.13 12-Bit DAC Linear Ramp (Pre-Layout)

Pin	Active State	Description	Input Stimulus	Period	Pulse Width	Delay	Edge Speed
Clk	High	Trigger DFF's	Vpulse - analogLib	1n	470p	20n	30p
X0	High	LSB	Vpulse – analogLib	2n	970p	31n	30p
X1	High	MSB – 10	Vpulse – analogLib	4n	1.97n	31n	30p
X2	High	MSB – 9	Vpulse – analogLib	8n	3.97n	31n	30p
X3	High	MSB – 8	Vpulse - analogLib	16n	7.97n	31n	30p
X4	High	MSB – 7	Vpulse – analogLib	32n	15.97n	31n	30p
X5	High	MSB – 6	Vpulse – analogLib	64n	31.97n	31n	30p
X6	High	MSB – 5	Vpulse – analogLib	128n	63.97n	31n	30p
X7	High	MSB – 4	Vpusle – analogLib	256n	127.97n	31n	30p
X8	High	MSB – 3	Vpulse – analogLib	512n	255.97n	31n	30p
X9	High	MSB – 2	Vpulse – analogLib	1.024u	511.97n	31n	30p
X10	High	MSB – 1	Vpulse – analogLib	2.048u	1.02397u	31n	30p
X11	High	MSB	Vpusle – analogLib	4.096u	2.04797u	31n	30p
Gate_Hi	NA	Bias for cascode	Current Reference	NA	NA	NA	NA
Gate_Lo	NA	Bias for cascode	Current Reference	NA	NA	NA	NA
Vdd	NA	DC Power (1 V)	Vpulse - analogLib	500m	0	20p	1n
Vss	NA	Ground (0 V)	Gnd – analogLib	NA	NA	NA	NA

Table 33 *– Input pins and stimuli for the 12-bit DAC simulation*

Pin	Description	Load on Pin	Load Value
I_Out	Complementary current output	Res – analogLib	10 Ohms
I_Out_Bar	Complementary current output	Res - analogLib	10 Ohms

Table 34 *– Output pins and their respective loads*

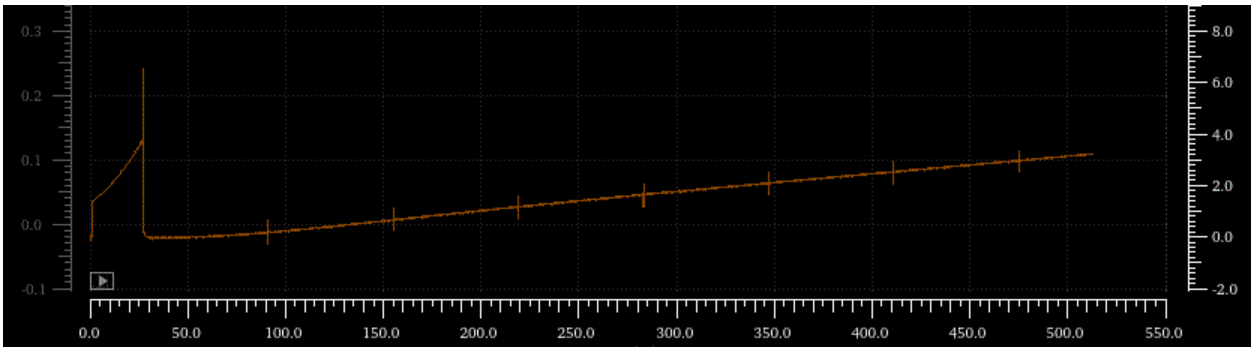


Figure 59 *– The output current for the linear ramp simulation*

Figure 59 shows 500ns of what was to be a much longer simulation. Its intent was to capture a complete linear ramp of all 4096 current cells contained in the 12-bit DAC. Unfortunately, the Linux server gave out at some point during the simulation and this was all that had been documented of the simulation. Note the transient current spikes – these occur on the rising edge of X5, which indicates that a row has been turned on. This is similar to the 8-bit post-layout simulation contained in 5.10.

In this simulation, as in the simulation from 5.12, the load at each complementary current output was set at 10 Ω in order to allow 30mA of total current flow with a 1 V power supply voltage.

Chapter 6 - Lessons Learned, Discussion, & Future Plans

6.1 Lessons Learned

Design, layout, and simulation of digital-to-analog converters presents a substantial challenge, and there was a significant learning curve involved with all three of those components.

The greatest design challenge faced was undoubtedly configuring the hierarchical nature of the DAC in such a way that mitigated the potential for human error. Recall that a 12-bit DAC with a fully segmented current steering topology contains 4096 cells arrayed in a 64x64 matrix. At the schematic level, a hierarchical method of 'copy-paste' was employed where-by a 2x2 matrix was wired together – then expanded to a 2x4 matrix, a 4x4 matrix, a 4x8 matrix, an 8x8 matrix and so on until a 64x64 was achieved. This was double and triple checked at every stage of 'copy and paste' to ensure that the schematic was error free – one can imagine attempting to debug a 64x64 array of cells in order to find one or two errant connections – akin to attempting to find a needle in a haystack.

From a layout standpoint, steps were again taken to ensure that the potential for human error was neutralized at every stage of the process. For example, all routing between cells in the DAC matrix was done at the lowest level of layout hierarchy such that the 'Mosaic' tool could be used in the analog layout to automatically create a 64x64 array – with all connections in place. Extremely fine measurements were taken during layout of the current cells to ensure that everything would connect perfectly when the 'Mosaic' function was engaged. Careful planning up front certainly saved time down the line.

A similar approach was taken for the D-Flip-Flops and decoders. The power and ground traces of each of these designs were routed such that they would perfectly connect to the DAC

array when married together. Additionally, the D-Flip-Flop was designed to be the exact same dimension as the respective current cell it would abut in the final layout. The author cannot stress enough the importance of planning up-front, as well as obsessive attention to detail when measuring and routing traces during the layout stage of an integrated circuit design project. When working with a design of this size and complexity, there is no other option – the hours spent measuring, measuring again, and even measuring a third time in some instances, will ensure that one doesn't have to start over completely on a layout due to errors that can't be resolved through debugging.

Another steep learning curve the author had to endure involved the layout vs. schematic tool. Again, attention to detail is the key here. In order to pass a layout vs. schematic run as one moves upward through different levels of hierarchy, special attention needs to be paid to the labeling of input and output nodes at each stage, else the layout tool will be unable to interpret between each level – this caused issues for the author early on in the layout stage, but was quickly resolved.

These challenges were relatively mild in comparison to those that arose during simulation and verification. Many approaches were tried – some were successful – many ended in total failure. One issue worth noting before delving into the full discussion of DAC simulation is the immense amount of time that each simulation took to run. The research team runs the IC design tool through a virtual Linux machine, with limited memory capabilities. This, coupled with the immense size that is part-and-parcel of fully segmented DAC's, contributed to the issue. Simulations taking two days or more were not uncommon. This made simulation debugging a slow process – only one simulation could be run at once, and often the author had to wait a day

or more before analyzing the results and assessing whether a change in approach was necessitated.

The following discussion will speak specifically to what worked for the K-State research team in question. Due to the manner in which the analog design tools were used by the team, as well as the limited resources available (memory, etc.), these are the approaches which proved successful for the unique situation the author found himself in. Due to the author's ignorance of the resources another research team, or industry professional may have at their disposal, it is only hoped that this discussion can serve as a resource for those struggling through the same issues our team faced – it is certainly not a 'one-size-fits-all' solution.

Some context may be useful before browbeating the reader with problems and their respective solutions, however. The first DAC simulations were conducted on a 4-bit fully segmented DAC. This was constructed from the same basic building blocks as the 12-bit version. It was to everyone's relief when this simulated correctly – the results can be seen in Figures 46 and 47. Of course, this was a pre-layout, or schematic, transient simulation and it completed in under an hour.

Emboldened by the success of a small design, the next step was to simulate an 8-bit DAC – fully segmented of course. Again, this was constructed from the same building blocks as the 4-bit designs and the eventual 12-bit design. Trouble quickly arose when attempting to simulate the 8-bit DAC, however.

The first issue involved discrepancies in the hierarchy of testbench simulations. Notice in all simulations from 5.8 – 5.13, the reference was set outside of the main symbol which contained the DAC current cells. The testbench was not configured this way by chance – attempts were made to run simulations with the current reference set in the same level of

hierarchy as the DAC cells. All of these simulations failed. Results were confusing to say the least – D-Flip-Flops were shown to exhibit characteristics of metastability, while all other digital logic failed to respond to any input stimuli.

The solution was to set the current reference outside the symbol in the testbench schematic. The reason for failure if this criterion was not met in unknown, but certainly worth investigating further in the future. There seemed to be convergence issues in the testbench schematic which repeatedly failed. For example, the analog simulator makes several guesses when attempting to converge – these guesses are listed in the transcript window when a simulation is running. Successful simulations seemed to always converge with 'gmin = homotopy'. If the simulator fails to find a solution with 'gmin = homotopy', it would then attempt 'gmin = source', and then 'gmin = dptran'. Simulations that failed would converge with 'gmin = dptran'. Again, it is not fully understood by the research team why these discrepancies in converge existed between the two designs, or why a discrepancy in an initial guess would cause such drastic alterations in simulation results.

Once simulations on the 8-bit design were converging properly, and showing promising results, efforts turned to reducing the inordinately lengthy simulation times for the 8-bit DAC. This proved tricky, and there were more failures than successes, although some progress was made.

The first method proposed to improve simulation time was to run the analog simulator in 64-bit mode. When this mode of operation was engaged, however, simulations suddenly began to fail. It is not clear why this was occurring, but certainly worth noting. This approach was taken in conjunction with multithreading. Multithreading didn't show a measurable difference in convergence or simulation time, so both approaches were abandoned. It should be noted that the

research team had access to four ‘threads’ – it is possible that if more threads had been present a more drastic difference would have been observed.

A second approach was to add resets to the D-Flip-Flops. It was proposed that convergence times were lengthy due to the simulator having to establish initial conditions for bi-stable circuits. The designs that were sent out for fabrication did not have an asynchronous reset functionality, but a test case was created to establish whether this was truly the issue, and schematic level simulations were run.

The D-Flip-Flops with asynchronous resets certainly allowed for faster convergence, and in hindsight, asynchronous resets should have been included in the realized designs. This is likely to be added in future revisions of the Ag-Radar chip. It is a simple alteration to the current D-Flip-Flop topology – it simply involves replacing two inverters with NAND gates.

The next approach was to use a pulse source for DC power in the testbench schematic – setting the value to 0 V at time zero, and then ramping up to the 1 V DC value used for Vdd sometime thereafter. This allows for a DC solution to converge in two iterations. Ultimately, this proved to make the most substantial difference in convergence time. The author has since learned that ramping Vdd is standard procedure when simulating large designs such as the DAC, but due to inexperience, was unaware of this approach early in the verification process.

One mistake that was made when taking this approach was to use a rise time for the pulse source in the range of 30 – 50 ps. This can cause other convergence problems, as such a rapid rise in DC supply voltage is difficult for the simulator to parse, and thus lengthens simulation time, or leads to failed simulation altogether. The solution is to use a rise time on the order of 1 – 10 ns. This is slow enough to ensure that issues which arose with the 30 – 50 ps rise time are mitigated.

6.2 Future Plans

On May 8th, 2020 the test-chip shown in Figure 34 was sent out for fabrication, with various submodules that will ultimately come together to form the K-State Ag-Radar. This test-chip will be returned to Kansas State in the Fall of 2020, and of course, will be tested rigorously such that the functionality of the submodules can be verified, and improvements can be made for the next tapeout date the research team is targeting.

A PCB is currently in development to facilitate testing of the chip, and it will allow for full-scale verification of all four submodules. An FGPA will be used to test the two DACs, and this will be placed directly on the board. A microcontroller will allow for communication between the user and the rest of the board (FPGA, LCD screen, and digital potentiometers) via an SPI protocol. The built-in ADCs on the microcontroller will serve to monitor the DC currents flowing to each submodule of the chip.

The layout of the board is shown in Figure 60. Figure 61 contains a '3-D' view of the board, which attempts to approximate the appearance of the board upon fabrication and assembly.

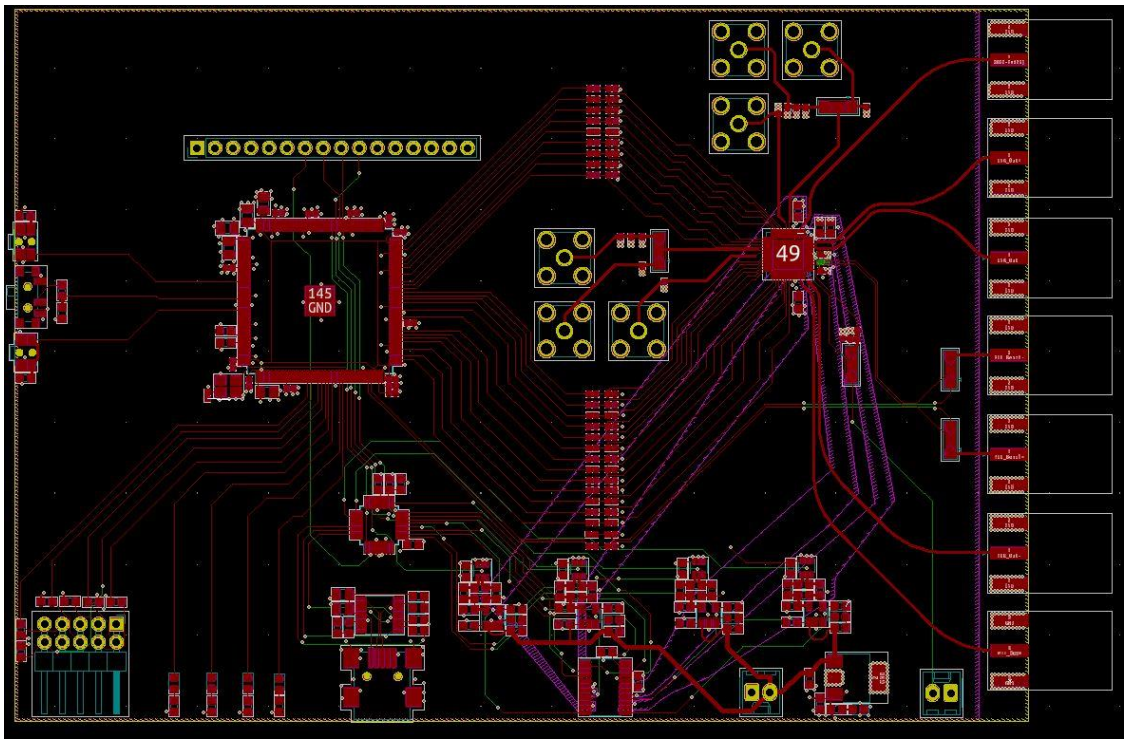


Figure 60 *– The test-board*

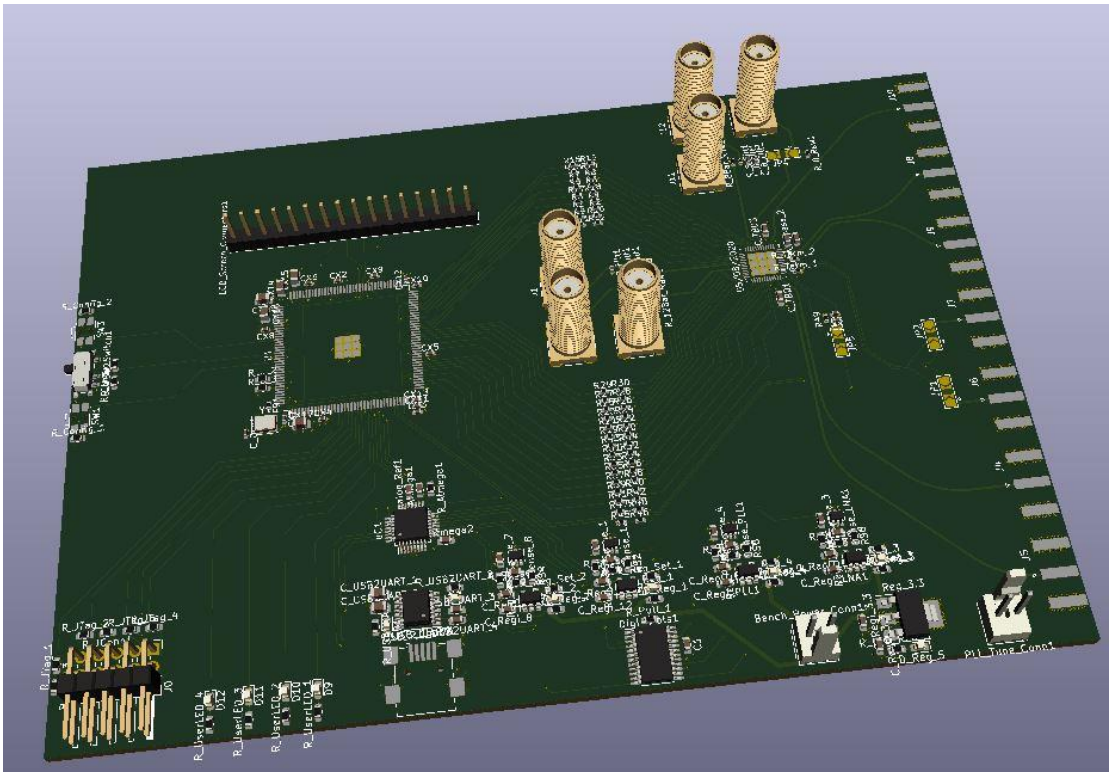


Figure 61 *– A 3-D view of the test-board*

It is the hope of the research team that digital libraries will be available to Kansas State as soon as possible. This would certainly be a huge stepping stone towards full realization of the radar on chip, allowing the DDS circuits discussed earlier in this thesis to be included on-chip. Certainly, alterations – improvements or updates – will be made to the Verilog scripts, but once a design flow is configured with the digital design tools, synthesis on-chip shouldn't be far off.

References

[1] R. A. Schulz, "Monolithic current-switch DAC improvements," in *IEEE Journal of Solid-State Circuits*, vol. 11, no. 2, pp. 338-341, April 1976

[2] W. Luschnig, R. Petschacher and E. Navratil, "Circuit Technique for Ultra Fast D/A Converters," *ESSCIRC '83: Ninth European Solid-State Circuits Conference*, Lausanne, Switzerland, 1983, pp. 61-64.

[3] S. Urquhart and P. H. Saul, "A 12-bit monolithic 70 ns DAC," in *IEEE Journal of Solid-State Circuits*, vol. 18, no. 3, pp. 302-305, June 1983

[4] P. H. Saul and J. S. Urquhart, "Techniques and technology for high-speed D—A conversion," in *IEEE Transactions on Electron Devices*, vol. 31, no. 2, pp. 196-202, Feb. 1984

[5] Chi-Hung Lin and K. Bult, "A 10-b, 500-MSample/s CMOS DAC in 0.6 mm/sup 2/," in *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 1948-1958, Dec. 1998

[6] J. Bastos, A. M. Marques, M. S. J. Steyaert and W. Sansen, "A 12-bit intrinsic accuracy high-speed CMOS DAC," in *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 1959-1969, Dec. 1998

[7] J. Deveugele and M. S. J. Steyaert, "A 10-bit 250-MS/s binary-weighted current-steering DAC," in *IEEE Journal of Solid-State Circuits*, vol. 41, no. 2, pp. 320-329, Feb. 2006

[8] C. Lin *et al*., "A 12 bit 2.9 GS/s DAC With IM3 $ ≪ -$60 dBc Beyond 1 GHz in 65 nm CMOS," in *IEEE Journal of Solid-State Circuits*, vol. 44, no. 12, pp. 3285-3293, Dec. 2009

[9] W. Tseng, C. Fan and J. Wu, "A 12-Bit 1.25-GS/s DAC in 90 nm CMOS With $ > $70 dB SFDR up to 500 MHz," in *IEEE Journal of Solid-State Circuits*, vol. 46, no. 12, pp. 2845-2856, Dec. 2011

[10] W. Lin and T. Kuo, "A low-spurious low-power 12-bit 300MS/s DAC with 0.1mm2 in 0.18μm CMOS process," *2013 IEEE International Conference of Electron Devices and Solid-state Circuits*, Hong Kong, 2013, pp. 1-2

[11] Shiyu Su, Tu-I Tsai, P. Sharma and M. S. Chen, "A 12-bit hybrid DAC with 8GS/s unrolled pipeline delta-sigma modulator achieving >75dB SFDR over 500MHz in 65nm CMOS," *2014 Symposium on VLSI Circuits Digest of Technical Papers*, Honolulu, HI, 2014, pp. 1-2

[12] K. Chang and C. Hsieh, "A 12 bit 150 MS/s 1.5 mW SAR ADC with adaptive radix DAC in 40 nm CMOS," *2016 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Toyama, 2016, pp. 157-160

[13] O. Aiello, P. Crovetti and M. Alioto, "Standard Cell-Based Ultra-Compact DACs in 40-nm CMOS," in *IEEE Access*, vol. 7, pp. 126479-126488, 2019

[14] D. Wang *et al*., "A 3GS/s 12-bit Current-Steering Digital-to-Analog Converter (DAC) in 55 nm CMOS Technology," in *MDPI Electronics*, April 2019

[15] Q. Ye, Y. Zhang, X. Li and Y. Zhang, "A 12-bit 10GSps ultra high speed DAC in InP HBT technology," *2017 2nd IEEE International Conference on Integrated Circuits and Microsystems (ICICM)*, Nanjing, 2017, pp. 9-13

[16] J. Kim *et al*., "Design and Analysis of a 12-b Current-Steering DAC in a 14-nm FinFET Technology for 2G/3G/4G Cellular Applications," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 10, pp. 3723-3732, Oct. 2019

[17] G. Peterson and W. Kuhn, "Coherent calibration cancellation for stepped frequency radar systems," *2018 IEEE Radar Conference (RadarConf18)*, Oklahoma City, OK, 2018, pp. 1594-1597

[18] E. McCune, "Time-filtered squarewave output from direct digital synthesis," *2010 IEEE MTT-S International Microwave Symposium*, Anaheim, CA, 2010, pp. 996-999

[19] D. E. Calbaza and Y. Savaria, "Jitter model of direct digital synthesis clock generators," *1999 IEEE International Symposium on Circuits and Systems (ISCAS)*, Orlando, FL, 1999, pp. 1-4

[20] A. L. Bramble, "Direct Digital Frequency Synthesis," *Thirty Fifth Annual Frequency Control Symposium*, Philadelphia, Pennsylvania, USA, 1981, pp. 406-414

[21] C. E. Wheatley and D. E. Phillips, "Spurious Suppression in Direct Digital Synthesizers," *Thirty Fifth Annual Frequency Control Symposium*, Philadelphia, Pennsylvania, USA, 1981, pp. 428-435

[22] P. H. Saul and M. S. J. Mudd, "A Direct Digital Synthesiser with 100MHz Output Capability," *ESSCIRC '87: 13th European Solid-State Circuits Conference*, Taunus-Tagungs-Zentrum, F.R. Germany, 1987, pp. 27-30

[23] C. G. Ekroot and S. I. Long, "A GaAs 4-bit adder-accumulator circuit for direct digital synthesis," in *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 573-580, April 1988

[24] H. T. Nicholas and H. Samueli, "A 150-MHz direct digital frequency synthesizer in 1.25- mu m CMOS with -90-dBc spurious performance," in *IEEE Journal of Solid-State Circuits*, vol. 26, no. 12, pp. 1959-1969, Dec. 1991

[25] X. Geng, F. F. Dai, J. D. Irwin and R. C. Jaeger, "24-Bit 5.0 GHz Direct Digital Synthesizer RFIC With Direct Digital Modulations in 0.13 μ m SiGe BiCMOS Technology," in *IEEE Journal of Solid-State Circuits*, vol. 45, no. 5, pp. 944-954, May 2010

[26] A. M. Ridwan, A. Faroqi, H. Nusantara and A. Munir, "DDS-Based 13.56MHz Sine Wave Generator for Wireless Charging System," *2018 International Conference on Applied Electromagnetics, Signal Processing and Communication (AESPC)*, Bhubaneswar, India, 2018, pp. 1-4

[27] F. Burcea, H. Habal and H. E. Graeb, "A New Chessboard Placement and Sizing Method for Capacitors in a Charge-Scaling DAC by Worst-Case Analysis of Nonlinearity," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 9, pp. 1397-1410, Sept. 2016