2011

Direct Digital Frequency Synthesizer Architecture for Wireless Communication in 90 NM CMOS Technology

Tri Trong Nguyen

Wright State University

DIRECT DIGITAL FREQUENCY SYNTHESIZER ARCHITECTURE FOR WIRELESS COMMUNICATION IN 90 NM CMOS TECHNOLOGY

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering

By

TRI TRONG NGUYEN
B.S., UNIVERSITY OF CINCINNATI, 2005

2011
Wright State University

WRIGHT STATE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

March 11, 2011

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Tri Trong Nguyen ENTITLED Direct Digital Frequency Synthesizer Architecture for Wireless Communication In 90 nm CMOS Technology BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering

Saiyu Ren, Ph.D.
Thesis Director

Kefu Xue, Ph.D.,
Chair
Department of
Electrical Engineering
College of Engineering and Computer Science

Committee on
Final Examination

Saiyu Ren, Ph.D.

Raymond E. Siferd, Ph.D.

Marian Kazimierczuk, Ph.D.

Andrew Hsu, Ph.D.
Dean, School of Graduate Studies

ABSTRACT

Nguyen, Tri Trong. M.S.Egr., Department of Electrical Engineering, Wright State University, 2011.
Direct Digital Synthesizer Architecture for Wireless Communication in 90 nm CMOS Technology.

Software radio is one promising field that can meet the demands for low cost, low power, and high speed electronic devices for wireless communication. At the heart of software radio is a programmable oscillator called a Direct Digital Synthesizer (DDS).

DDS has the capabilities of rapid frequency hopping by digital software control while operating at very high frequencies and having sub-hertz resolution. Nevertheless, the digital-to-analog converter (DAC) and the read-only-memory (ROM) look-up table, building blocks of the DDS, prevent the DDS to be used in wireless communication because they introduce errors and noises to the DDS and their performances deteriorate at high speed. The DAC and ROM are replaced in this thesis by analog active filters that convert the square wave output of the phase accumulator directly into a sine wave. The proposed architecture operates with a reference clock of 9.09 GHz and can be fully-integrated in 90 nm CMOS technology.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

ACKNOWLEDGMENT

First of all, I would like to thank my adviser and thesis director, Dr. Saiyu Ren for providing me with invaluable guidance on my research. Throughout my work on the thesis, Dr. Ren has opened my mind to new fascinating knowledge.

I'm sincerely grateful for her immense patience and dedication in helping me developing my research and analytical skills that will be tremendously valuable in my career as an electrical engineer. Always, she offers me solutions to problems I encountered with encouragements and advices.

I am also grateful to Professor Raymond Siferd, who introduced me to the area of RF Analog CMOS, and kindly served as my committee member.

It has been a privilege for me to learn and grow in the Electrical Engineering department of Wright State University.

I. INTRODUCTION

Motivation

In the last few years, there has been a huge evolution in wireless communication technologies: in both the mobile technology and the Wireless Local Area Network (WLAN) technology. With the emergence of the 4G wireless standard, wireless communication technology holds the promise of worldwide roaming using a single hand-held device to satisfy all professional and personal networking needs.

This recent boom in wireless application has driven up the demand for inexpensive, fully-integrated, low power, high performance transceivers.

Typically, Gallium Arsenide (GaAs) and Silicon Germanium (SiGe) technologies used to dominate in high speed application with high costs and high power consumption. From a cost standpoint, CMOS technology is the least expensive due to its higher production yield and its higher density of on-chip integrated circuits. Reducing the number of off-chip components is necessary to lower cost and size of circuit card assemblies in a transceiver, CMOS technology is the preferred choice for integrating RF front-end.

With the constant decrease in transistor's size in the deep-sub micron, it is possible for CMOS technology to support high-end RF circuits. Combination of design techniques such as device scaling (thinner oxide), shortening of the transistor's channel-length, lowering of the voltage supply helps in attaining higher frequencies of operation and low power consumption.

In the front-end circuit of wireless transceivers, channel selection and frequency translation are carried out by frequency synthesizers. This thesis focuses on the design of one type of frequency synthesizer for use in wireless communication: the direct digital synthesizer (DDS).

Contributions of this Work

This work aims at upgrading the DDS technology in such a way that Direct Digital Synthesis can be used in a wide range of wireless applications instead of a being constrained to applications in specific areas.

The objective can be met by reviewing the DDS architecture and improving it. This thesis attempts at replacing the phase to amplitude converter portion of the traditional DDS with analog active filters that operate at higher frequencies, consume less power, and generate less noise than the ROM Look-up table and Digital to Analog Converter used in the traditional DDS.

This work also suggests a design of high speed full-adders and digital optimization techniques that boost the operating speed of the phase accumulator allowing it to be used in faster DDS.

Organization of Thesis

The thesis is organized into seven chapters. This introductory chapter examines different types of popular frequency synthesizers widely used in various applications. The operating concept and characteristics of each type are pointed out in order to compare the advantages and disadvantages of each type. The theory of operation of the traditional DDS is also explained in this part. The characteristics that make the DDS attractive in wireless applications and the ones that limit its usage are described.

The second chapter starts with the proposition of a new DDS architecture suitable for high speed wireless applications. The new DDS architecture requires new building blocks that are covered in the rest of this chapter. A design of a high-speed full-adder cell capable of operating at 10 GHz is proposed. The adder design is intended for use in the phase accumulator that makes up the

DDS. Delay optimization, parallelism, and combination of static logic with pass-gate logic allow the full-adder to operate at high speed using 90 nm CMOS technology without having to recur to the weak turn-on mode operation.

Next, a design of the phase accumulator that renders the DDS programmable is presented. Pipelining and delay reducing techniques are employed to allow the operating speed of the DDS be independent from the number of bits accuracy of the DDS.

In order to convert the output of the phase accumulator into a sine wave, active filters are used. The conversion of filtering consists simply of stripping harmonic frequencies from the square wave output of the phase accumulator. The active filter’s configuration is a 3 stages stagger-tuning cascode amplifier. The design takes advantages of the integrations of on-chip passive component to achieve high speed operation.

The last function block is the design of the output driver that serves as an analog buffer between the active filter and the external load. The design is based on a class-A power amplifier to achieve the best linearity possible.

The simulation results of the DDS are in chapter 3. An overview of the results is laid out in chapter 4 to summarize and judge the accomplishments of the results versus the objectives. Suggestions for future work, in chapter 5, are listed to further expand the DDS capabilities.

Chapter 6 contains references that are cited in this thesis. The attachment in the last chapter contains VHDL code for an 8-bit phase accumulator in an FPGA. This code serves to demonstrate the capability of the pipelining method applied to large arithmetic logic.

Frequency Synthesizers

Definition and characteristics

The purpose of a frequency synthesizer (FS) is to produce one or multiple requested frequency outputs from one reference frequency source and a control input.

The quality and performance of an FS is characterized mainly by its bandwidth, its frequency resolution and finally its frequency purity. A trade-off of these qualities and performances is made in choosing the appropriate type of FS.

Ideally, the output of an FS is a pure sinusoidal waveform. In reality phase noises and spurs exist near the desired carrier frequency. The phase noise is measured in dBc/Hz, a ratio of phase noise in 1 Hz bandwidth at the offset frequency. [1]

Implementation of Frequency Synthesizers

Frequency synthesizers are classified into four general groups:

1. Direct analog synthesizer (DAS)
2. Direct Digital Synthesizer (DDS)
3. Phase-locked Loop Frequency Synthesizer (PLL-FS)
4. Delay-locked Loop Frequency Synthesizer (DLL-FS)

Table 1 briefly describes the different characteristics and application of each of the FS type.

Table 1: Classification of Frequency Synthesizers [1]

Direct Synthesis	DAS	Multiplier + Mixer + Divider + Band-pass Filter	
	DDS	Phase Accumulator + DAC	
Indirect Synthesis	PLL-FS	Integer-N	
		Factional-N	Phase Estimation by DAC
			Random jittering
			Noise shaping by ΣΔ
			Phase Interpolation
			Pulse Generation
	DLL-FS	Frequency multiplied by number of equal-spacing phases	

Direct Analog Synthesizer

The Direct Analog Synthesizer (DAS) utilizes frequency multipliers, dividers, mixers and band-pass filter to obtain the desired synthesized frequency. The architecture of DAS is shown in Figure 1.

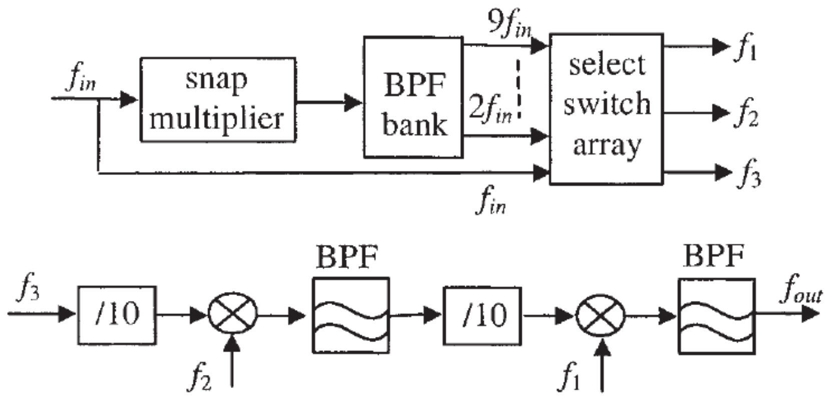


Figure 1: Architecture of DAS [1]

Multiple output frequencies can be generated from a single frequency source by repeating the circuits in cascading stages.[2] There are many advantages for using this type of synthesizers. The output signals are clean with excellent phase noise because they are derived directly from a single reference source.

DAS also offer the ability of rapidly switching the frequency. On the other hand, the limitations of DAS originate from its architecture that produces large size circuits that, in turn, demand large amount of power. Figure 1 shows an example of DAS. In this example the output frequency can be calculated as:

$$f_{out} = f_1 + 0.1 f_2 + 0.01 f_3 \qquad [1.1]$$

The frequency is controlled by the frequency division. In this case the frequency resolution is $0.01 f_{in}$.

Phase-locked Loop Frequency Synthesizer

Phase-locked Loop based Frequency Synthesizers are largely the most commonly used synthesizers. They have their own benefits and problems. Just as listed in Table 1, there are two types of PLL based frequency synthesizers: the Integer-N PLL-FS, and the Fractional-N PLL-FS. They are discussed in the following paragraphs.

Integer-N PLL-FS

The Integer-N PLL based frequency synthesizer (FS) consists, as shown in Figure 2, of a phase-frequency detector (PFD), a charge-pump (CP), a loop filter, a voltage-controlled oscillator (VCO), and a programmable frequency divider.

The output frequency of this type of FS is a multiple of the reference frequency:

$$f_{out} = N.f_{REF}\text{, where N is an integer} \qquad [1.2]$$

From this equation, the frequency resolution is equal to the reference frequency f_{REF}. To obtain high resolution in narrow-band applications, the reference frequency must be small and the frequency divide ratio must be large.[1]

The low reference frequency and narrow loop-bandwidth characteristics of the conventional integer-N PLL pose several problems:

- The lock time is long
- The reference spur and its harmonics are located at low offset frequencies
- The large divide ratio (N) causes the in-band phase noise to increase.
- The phase noise of the VCO can remain unacceptably high at low offset frequencies. [1]

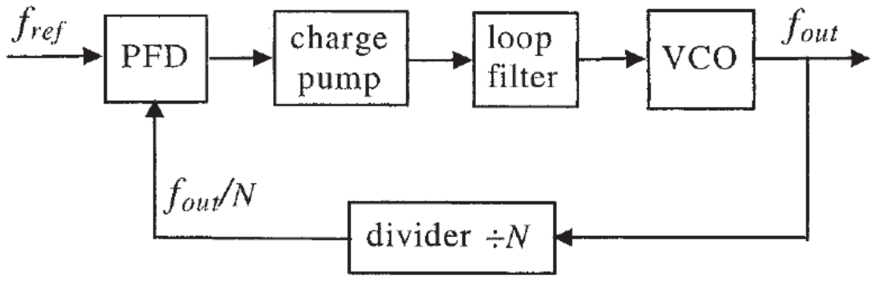


Figure 2: Integer-N PLL-FS [1]

Factional-N PLL-FS

To overcome the flaws of the conventional integer-N synthesizers, the Fractional-N frequency synthesizers were invented. Unlike the Integer-N FS, the fractional synthesizers support the frequency division ratio as fractions so that a larger reference frequency can be used to achieve better frequency resolution.

However, the main disadvantages of the fractional-N frequency synthesis are the unwanted low frequency spurs created by the dual-modulus divider. Spur Reduction Techniques, listed in Table 2, are needed to make the fractional-N FS work. An example of the architecture of a fractional-N FS using DAC phase estimation is shown in Figure 3. An accumulator is used to control the instantaneous divide ratio. If the overflow is equal to 1, the divide ratio is $N_B + 1$, otherwise it is just N_B.

Table 2: Fractional-N FS, Spurs Reduction Techniques [1]

Technique	Feature	Problem
DAC phase estimation	cancel spur by DAC	analog mismatch
Random jittering	randomize divide ratio	frequency jitter
ΣΔ noise shaping	modulate divide ratio	quantization noise
Phase interpolation	inherent fractional divider	interpolation jitter
Pulse generation	or multiphase VCO	interpolation jitter

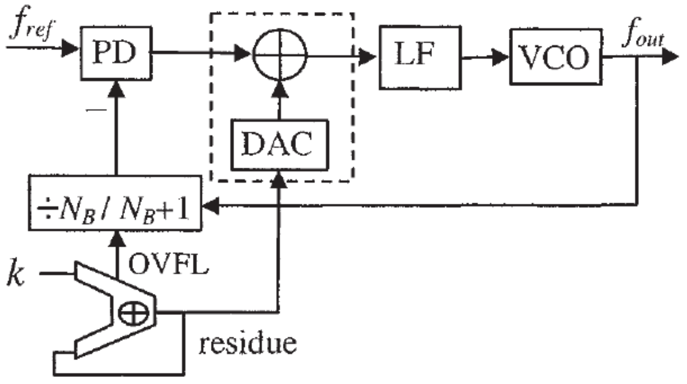


Figure 3: FN-FS using DAC phase interpolation [1]

Delay-Locked Loop (DLL) Frequency Synthesizer

Frequency synthesis can be accomplished with Delay-Locked-Loops. In general, DLLs are not the best choice for frequency synthesizers due to the fact that the frequency is not reprogrammable and that the delays can be hard to control. Rather, DLL are useful to de-skew clock signal, to multiply frequency or to generate multiphase signals. [3] The DLL-FS consists of a phase detector, a charge pump, a low pass filter, a voltage controlled delay line, and an edge combiner. The block diagram of a DLL-FS is shown in Figure 4.

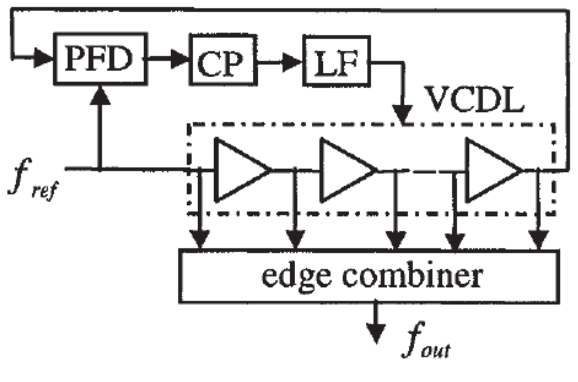


Figure 4: DLL-FS Architecture [1]

Direct Digital Synthesizer

Direct digital synthesis (DDS) is a technology that uses digital circuits to generate a programmable frequency output signal from a fixed-frequency clock source and a digital tuning word. In its simplest form, the DDS is formed by a phase accumulator and a phase to amplitude converter.[4] A digital phase accumulator is used to accumulate the phase angle of a sine wave, then, a decoder matches the phase to the corresponding amplitude contained in a look-up table (LUT). The digital tuning word is used to program the phase angle increment. Its width (number of bits) determines the tuning resolution of the output frequency.

The integration of DDS onto a single transceiver chip enabled this technology to be cost-competitive, high-performance, and small. Gradually, DDS are employed in a broader range of applications. In certain areas, the DDS is an attractive substitution to the traditional PLL synthesizers. The DDS has the following distinct advantages over other types of frequency synthesizers.

DDS Advantages [1]:

- Micro-Hertz frequency tuning resolution and sub-degree phase tuning capability

- Fast "hopping speed" when changing from one output frequency to another.
- No de-tuning or analog parameters degradation due to components aging
- Ease of digital control interface

Traditional DDS Theory of Operation

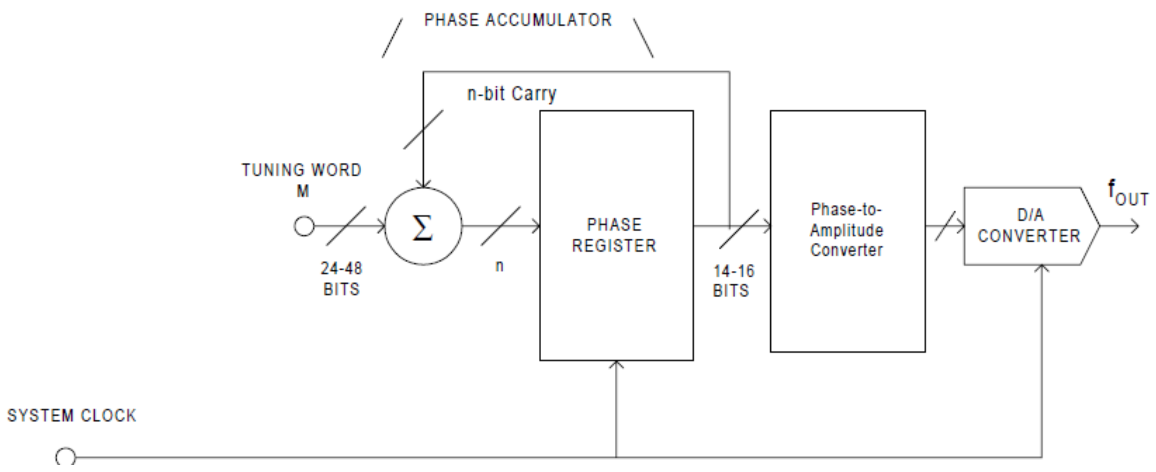


Figure 5: Simple DDS Block Diagram [4]

The architecture of the basic DDS, shown in Figure 5, is composed of an N-bit phase accumulator, a phase to amplitude converter and a digital converter. At the subsystem level, the phase accumulator consists of "N" adders and chains of registers. The phase to amplitude converter consists of a sine-wave amplitude look up table (LUT) and Digital to Analog Converter. [4]

The theory of operation of the basic DDS can be easily understood by digitalizing the amplitude of a sine wave into bits of a computer memory as shown in Figure 6. Each address of the memory contains the amplitude data at an angle in the 360° phase of a sine wave. When a DAC scans through the entire memory of the LUT, one address per clock interval, the output of the DAC forms a sine wave. The role of the phase accumulator is to increment the address of the LUT which corresponds to incrementing the angle. The incremental step in the address reading is set by the digital control input. The number of skipped addresses reflects the change in the frequency of the sine wave. To obtain a smoother sine wave a low-pass filter is necessary after the DAC.

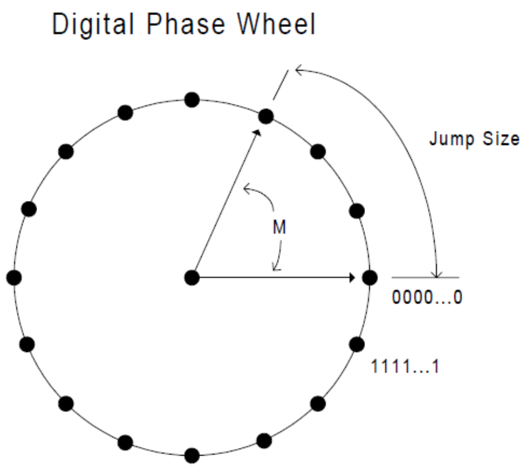


Figure 6: Sine Wave Digital phase wheel [4]

Another easy way to understand the basic operation of the DDS operation is through modeling a simplified 8-bit DDS in MATLAB. An example of MATLAB codes is shown in Figure 7.

In this model, a DDS with an 8-bit phase accumulator and a 5-bit address LUT is simulated. The digital control input, also called the frequency tuning word "M", is given as "4".

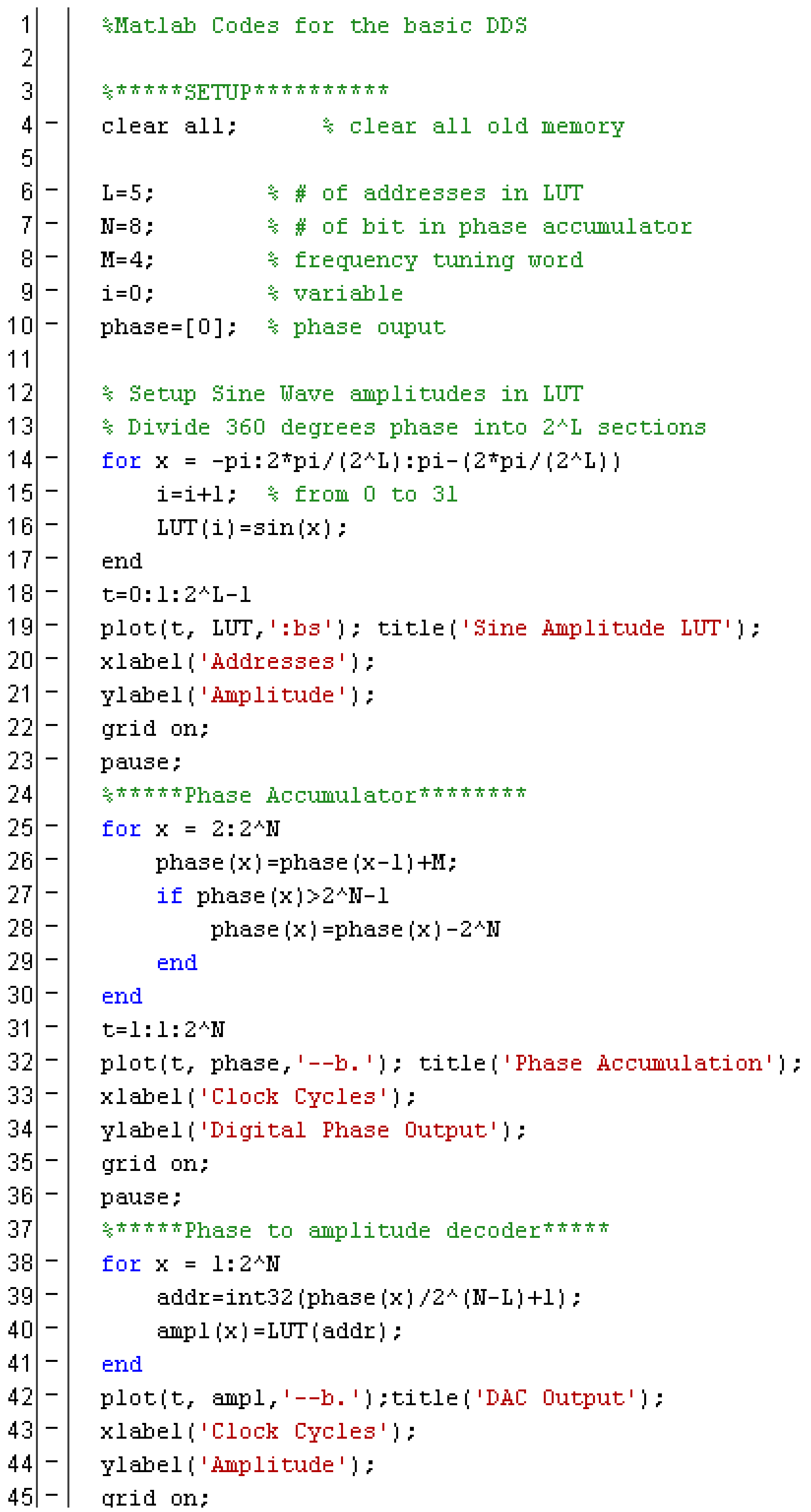
Setting M as the frequency tuning word, N as the N-bit phase accumulator, and f_C as the clock frequency, the output frequency of the DDS, f_O can be defined as:

$$f_O = \frac{M \times f_C}{2^N} \qquad [1.3]$$

"N" defines the frequency resolution of the DDS. A wider "N" bit phase accumulator can have sub-hertz frequency resolution, but also required a wider memory.

"M" programs the incremental step of the phase and equivalently the incremental step of the LUT addresses to read. The phase accumulator cycles back to the beginning address on overflow. The faster the phase accumulator overflows, the higher is the output frequency.

The first plot obtained from the MATLAB codes, shown in Figure 8, represents a scan through the "L-bit" wide memory of the LUT which contains 2^L numbers of data points. Each address contains a preset amplitude of the sine wave. The second plot in Figure 9 shows the phase accumulation, done by adding the frequency tuning word "M" to the previously accumulated phase at every clock tic.



```
%Matlab Codes for the basic DDS

%*****SETUP**********
clear all;      % clear all old memory

L=5;         % # of addresses in LUT
N=8;         % # of bit in phase accumulator
M=4;         % frequency tuning word
i=0;         % variable
phase=[0];   % phase ouput

% Setup Sine Wave amplitudes in LUT
% Divide 360 degrees phase into 2^L sections
for x = -pi:2*pi/(2^L):pi-(2*pi/(2^L))
    i=i+1;  % from 0 to 31
    LUT(i)=sin(x);
end
t=0:1:2^L-1
plot(t, LUT,':bs'); title('Sine Amplitude LUT');
xlabel('Addresses');
ylabel('Amplitude');
grid on;
pause;
%*****Phase Accumulator********
for x = 2:2^N
    phase(x)=phase(x-1)+M;
    if phase(x)>2^N-1
        phase(x)=phase(x)-2^N
    end
end
t=1:1:2^N
plot(t, phase,'--b.'); title('Phase Accumulation');
xlabel('Clock Cycles');
ylabel('Digital Phase Output');
grid on;
pause;
%*****Phase to amplitude decoder*****
for x = 1:2^N
    addr=int32(phase(x)/2^(N-L)+1);
    ampl(x)=LUT(addr);
end
plot(t, ampl,'--b.');title('DAC Output');
xlabel('Clock Cycles');
ylabel('Amplitude');
grid on;
```

Figure 7: DDS in MATLAB

Once the maximum phase is attained, the 8-bit phase accumulator discards the overflow and cycle back to the beginning. The digital output of the phase accumulator is used to match to the correct address of the LUT.

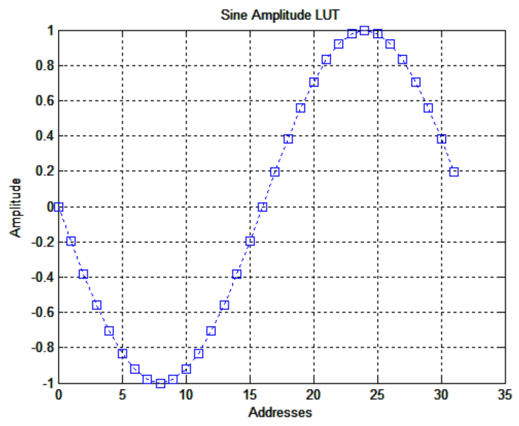


Figure 8: Sine wave amplitude LUT with 2^5 points

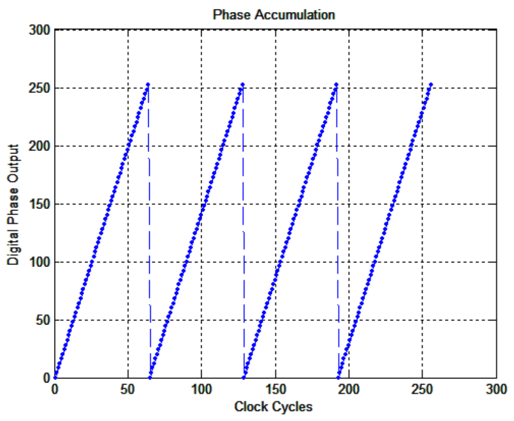


Figure 9: Digital Phase Out. of 8-bit accu. and M=4

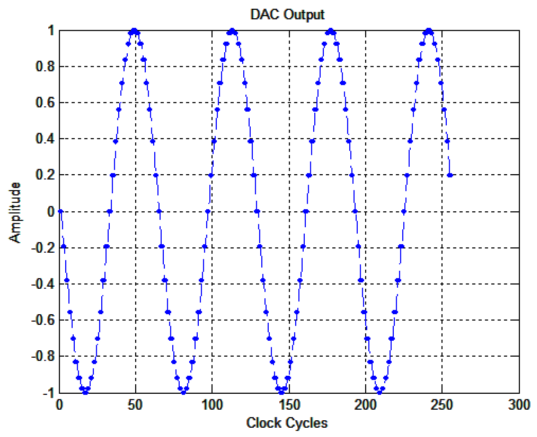


Figure 10: DDS Output with M=4

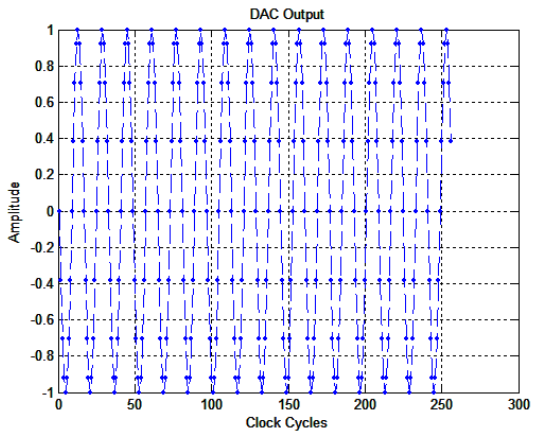


Figure 11: DDS Output with M=16

The plot, in Figure 10, shows the output of the DDS. Each point on the graph represents the sine wave amplitude data stored in the LUT. In this example, only 32 data points from the LUT are used to plot the sine wave. Figure

11 shows another output of the DDS with a frequency tuning word 4 times greater as previously. The output frequency becomes four times greater and the output sine wave uses 4 times less data points.

Limitations of the traditional DDS

For years, the DDS have stood in the shadow of PLLs for many reasons. They were known to consume a great amount of power while being a source of spurious noise energies. There is a performance trade-off in a DDS system. A wider operating bandwidth causes the power consumption and the spurious level to increase.[5]

Their capability of fast frequency hopping and superior frequency resolutions tuning have put them for use in very specific applications such as instrumentation, and military radar systems and communications: the basic DDS architecture is certainly not suitable for portable wireless communication without modification.

The main deficiency of the traditional DDS is in its architecture. Conceptualized many decades ago, the architecture of the DDS does not favor both high speed operation and low power consumption. Furthermore, the spurious generation behavior of the DDS complicates designs enormously. Various analysis methods were developed to allocate the locations of the spurs. Once the frequency locations of the spurs are foreseen, spurs management takes place to reduce the spurs to an acceptable level or to push the spurious line spectrum out of the operating bandwidth. As a result, most DDS applications are constrained to operate with only a fraction of the bandwidth.[1]

The DDS has six sources of noise and spurs: [6]

1. "The phase noise of the reference clock"
2. "The truncation of the phase accumulator bits addressing the sine ROM LUT"
3. "A distortion from compressing the sine ROM LUT"
4. "The finite precision of the sine samples stored in the ROM LUT"
5. "The DAC conversion"
6. "The post-filter error"

Alike any other digital systems, inevitably the spectral characteristics of the reference clock appear at the output of the DDS as a result of clock feed-through. However due to the frequency division, the noise generated by the reference clock is attenuated.[1] Low phase noise and low-spurs clocks are always preferred in DDS designs. Today's technology offers clock generator with jitters in the femto-second. For the reason that DDS devices work with sampled data, the Nyquist sampling theorem applies. The output frequency of a DDS can't exceed half of the reference clock frequency. Therefore, the reference clock frequency is out of the operating bandwidth of the DDS.

At the output of the DDS, another source of noise is from the reconstruction passive low-pass filter. This filter is needed to remove the high frequency sampling components of digital switching. Since the filter is passive, the amplitude response and delays are of concerns. [6]

- Phase Truncation Error

In application requiring fast frequency hopping in the sub-hertz resolution, a DDS system with a minimum of 32-bit phase accumulator and an 8-bit D/A could be conceptualized. This size of DDS demands for 4-gigabytes of memory in order to meet the resolution's demand. However, altogether, cost, size, power consumption, and operating speed forbid the use of such a memory size. The DDS designer is forced to reduce the size of the LUT. As result, several bits (LSB) from the outputs of the phase accumulator can be discarded or truncated. This reduced-size LUT leads to phase truncation error which contributes to noise.

The following examples show the effect of phase truncation when utilizing an 8-bit accumulator along side with a 5-bit LUT. The maximum total of 32 points is used to draw the sine wave. The memory's reduction went from

256 (2^N) addresses to 32 (2^L) addresses. When programmed with a frequency tuning word of 4, the amplitude reading of the DAC skips three addresses of the LUT at every clock. Per consequence, the DDS just use eight amplitudes out of 32 to construct the sine wave.

$$f_O = \frac{4 \times f_C}{2^8} = \frac{f_C}{64} \text{ , with M=4} \qquad [1.4]$$

The following figures illustrate the effects of phase truncation at the output of the DDS.

In the first example, in Figure 12 and Figure 13, a full sized LUT is simulated to serve as references. The output of the phase accumulator is not truncated. The output of the DAC shows a smooth sine wave.

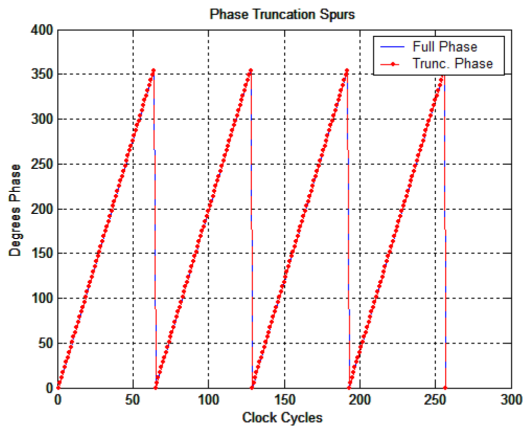


Figure 12: Digital Phase N=8, L=8, M=4

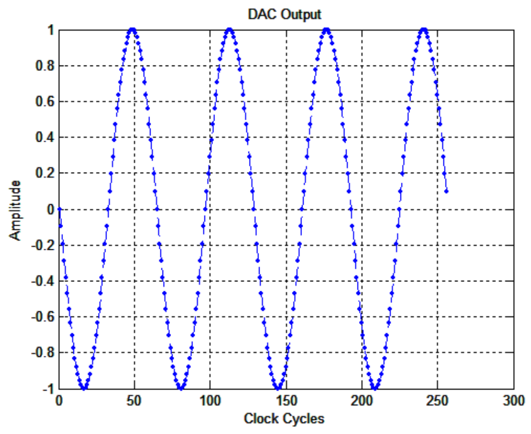
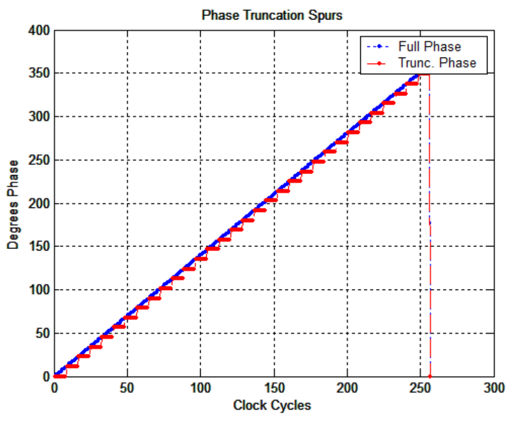


Figure 13: N=8, L=8, M=4, DDS Output

In the second example, the LUT is reduced by 3 bits; reducing from 8 bits (N) to 5 bits (L). In Figure 14, the output of the phase accumulator resembles a staircase.



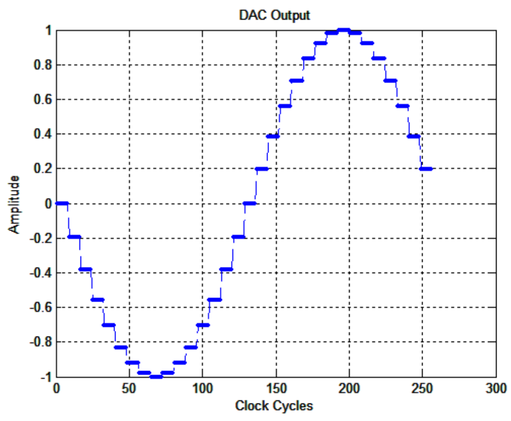
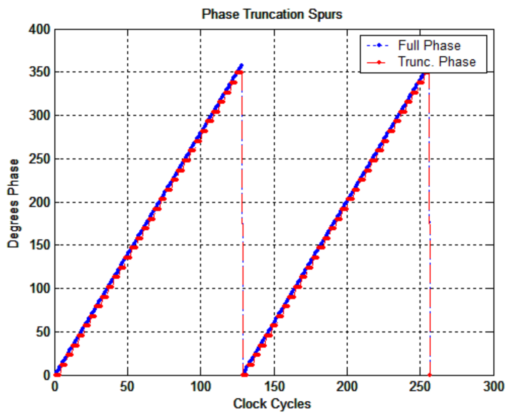


Figure 14: Digital Phase N=8, L=5, M=1

Figure 15: N=8, L=5, M=1, DDS Output

Since the lower bits are discarded (truncated), the phase accumulator repeats the same output for several clock cycles. The ending result at the output of the DAC is a less smooth sine waveform, shown in Figure 15.



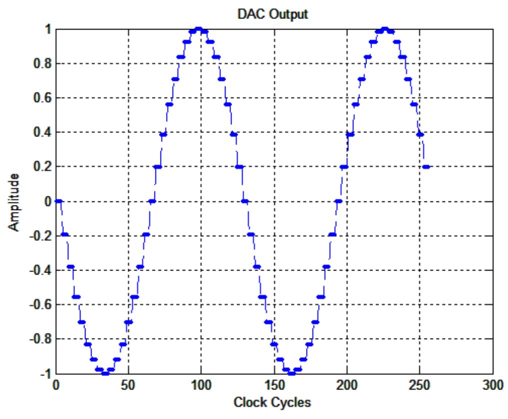


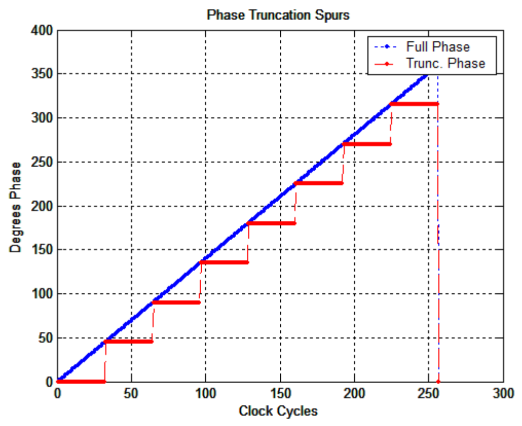
Figure 16: Digital Phase N=8, L=5, M=2

Figure 17: N=8, L=5, M=2, DDS Output

The tuning word is increased in the next example from 1 to 2. The phase accumulator increases the digital phase output faster. The same phase outputs are repeated less often after each clock cycle. However, the sine waveform remains distorted. If the tuning word is increased again, there is a value where the sine wave becomes smooth because the digital phase output is not repeated.

Via this MATLAB simulation, it can be observed that spurs resulting from phase truncation can vary drastically with the output frequency. However, the phase truncation sequence is periodic.

The acceptable reduction of the size of the LUT depends of the design of the low-pass filter used to smooth out the sine waveform.



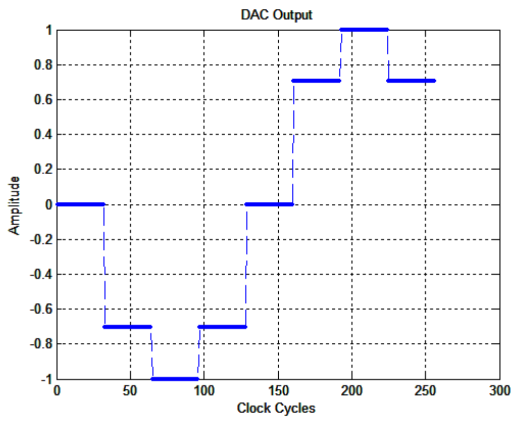


Figure 18: Digital Phase N=8, L=3, M=1

Figure 19: N=8, L=3, M=1, DDS Output

Figure 18 and Figure 19 show a highly distorted sine waveform that requires high amount of filtering.

- Phase to Amplitude Mapping Error: the finite precision of the sine samples stored in the ROM LUT

The finite precision in the sampled sine ROM amplitude stored in the LUT leads to error in the DDS output. Many algorithms were developed with the idea of reducing the mapping error. Even the complex algorithms have error if the memory size is limited. When the look-up table and DAC have finite resolution, spurs will be generated.[7]

An approximation of the amplitude of the sine wave is stored for every timed interval. The nearest value is chosen so that the stored amplitude corresponds to a bit-word the DAC can read. Figure 20 shows a mapping approximation of a sine wave. A good design rule requires the sine amplitude data to be 3-bit wider than the resolution of the DAC. [1]

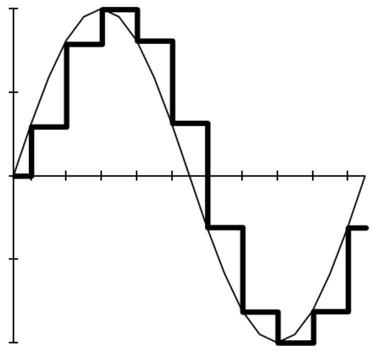


Figure 20: Sine Wave Mapping Approximation [7]

- Error of DAC

At high speed (>50 MHz) and high resolution applications (>10 bits), the DAC dominates as a source of noise and spurs in the DDS. [6]

Ideally, the DAC conversion occurs immediately without delay. Realistic DACs have a settling time that limits the operating frequency of the DAC. The internal circuit of the DAC has a propagation delay and the output driver has a limited slew rate.

Another error parameter of the DAC is the integral nonlinearity (INL). The integral nonlinearity describes the linearity of the transfer function of the DAC. It corresponds to the maximum deviation from an ideal output conversion. Preferably, a value of INL less than ±1 LSB is desirable in a DDS system.

Additionally, another type of nonlinearity affects the DAC’s output accuracy: it’s the differential nonlinearity error (DNL). The (DNL) corresponds to the difference between an actual step height and the ideal value of 1 LSB. Preferably, a value of DNL less than 1 LSB is desirable in a DDS system. [6]

II DDS ARCHITECTURE DESIGN

Proposed architecture

A successful DDS design designated for wireless communication must deviate from the traditional DDS architecture. The phase to amplitude converter has speed constrains, consumes relatively large amount of power, and introduces noise to the output.

The proposed architecture eliminates the ROM LUT and the DAC altogether. The phase to amplitude converter is replaced by active filter banks with the individual filters tuned to a fraction of the DDS bandwidth. The input to the filter bank comes directly from the phase accumulator. The role of the filter bank is to convert the input square wave into a sine wave by removing the harmonics content frequency from the digital square wave input. From the frequency tuning word information, a multiplexer connects the phase accumulator to the appropriate filter which operates at the correct bandwidth frequency. Following the filter, the sine wave output is connected to an output driver through a de-multiplexer. The architecture of the proposed design is shown in Figure 21.

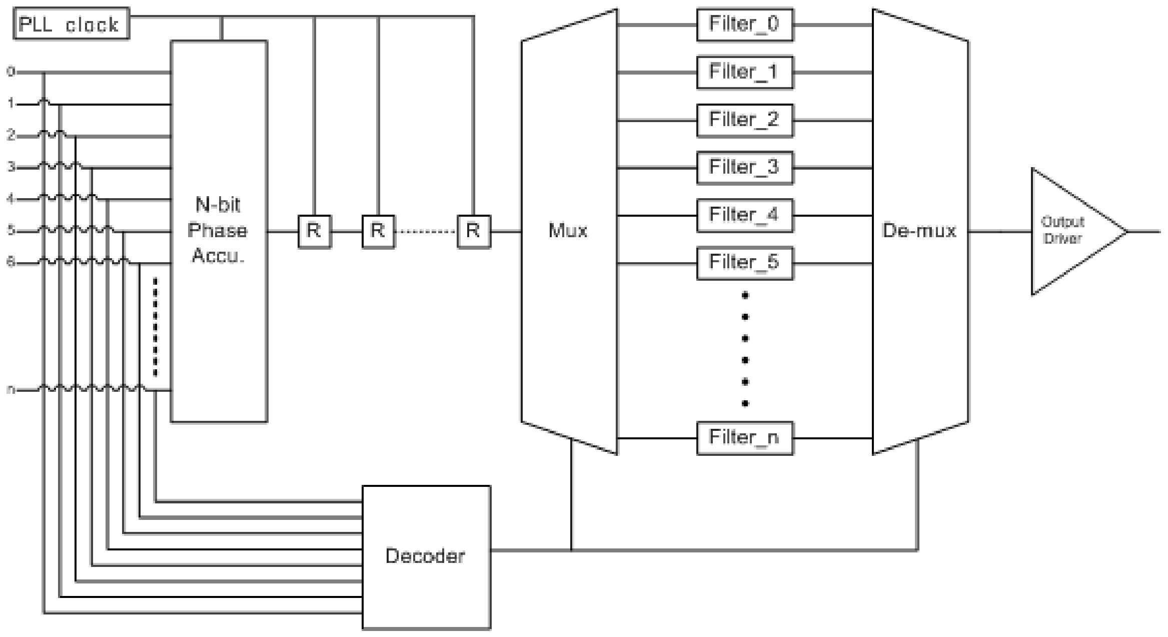


Figure 21: Proposed DDS Architecture

The role of the phase accumulator has changed from being an address counter to directly a frequency generator. The phase information is no longer needed since the ROM LUT is no longer used. With the analog filter's role of a sine wave generator, the new purpose of the phase accumulator is to generate a frequency.

As the frequency tuning word is accumulated inside the phase accumulator at every clock tic, the outputs of the phase accumulator changes state after a certain number of clock cycles. The frequency at which the outputs switch states can be controlled directly via the value of the frequency tuning word.

Once the frequency tuning word is entered, one output or a combination of outputs creates digitally the desired frequency under the form of a square wave output. The square wave output is connected to the analog filter banks via multiplexers. The multiplexer, de-multiplexer, and decoder are not covered in this thesis.

The analog filter bank, which is replacing the phase-to-amplitude- converter, operates as a Harmonics Stripper. As shown in Figure 22, filtering is used to convert the digital square wave output of the Phase Accumulator into a sine wave.

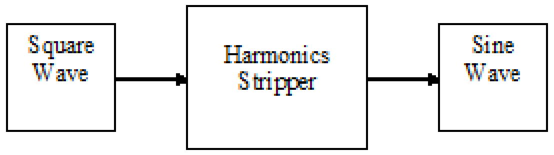


Figure 22: Sine Wave Generator

The last building block in the proposed architecture is the output driver. It is used as a buffer between the de-multiplexer and external load of the DDS.

Fast Adder

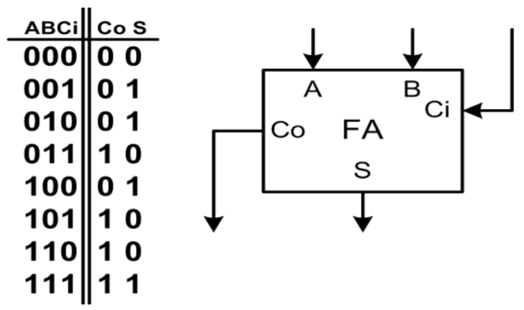
Full-adders are standard cells found in every digital processing application. Beside the standard adders found in most libraries, many different designs are published in countless papers to obtain different performances. Certain designs focus on reducing transition activity and recycling charge to obtain low-power of operation because switching activity and node capacitances affects the power dissipation.[9] Other designs have low short-current power consumption because their logic is not based on the static CMOS. [8] Large arithmetic circuits requiring high number of adders may favor adders with low number of transistors count.

For this DDS, the propagation delay of an adder is critical. The delay of an adder depends generally on three characteristics of the circuit: first, the number of gates in the critical path, second, the number of transistors in series in the pull up/down path, and last, the transistor sizes. Bubble logic can reduce the number of inversion levels in a ripple adder.

The performance of the adders is influenced by the logic style of different circuit families; Static CMOS, Transmission Gates, Complementary Pass Transistor Logic (CPL), Cascode Voltage Switch Logic (CVSL)… Each family has its own advantages and limitations. [10]

The objective of the adder design is to build an adder in 90nm CMOS technology, capable of operating at 10 GHz, but, without using the depletion mode, the weak turn on mode, or the current mode.

All full adders perform the same function that can be described by the truth table in Figure 23.



ABCi	Co S
000	0 0
001	0 1
010	0 1
011	1 0
100	0 1
101	1 0
110	1 0
111	1 1

Figure 23: Full-adder function

<u>Mirror Adder</u>

Shown in Figure 24 is the most popular full-adder used in digital circuits, the mirror adder. It’s built from 28 transistors, of which, 4 transistors are used to invert the SUM and Carry-out (COUT) outputs. In a ripple adder configuration with even number of bits, inversion property can be employed to the COUT bit to save two transistors per adder. And at the same time, the carryout propagation delay is reduced.

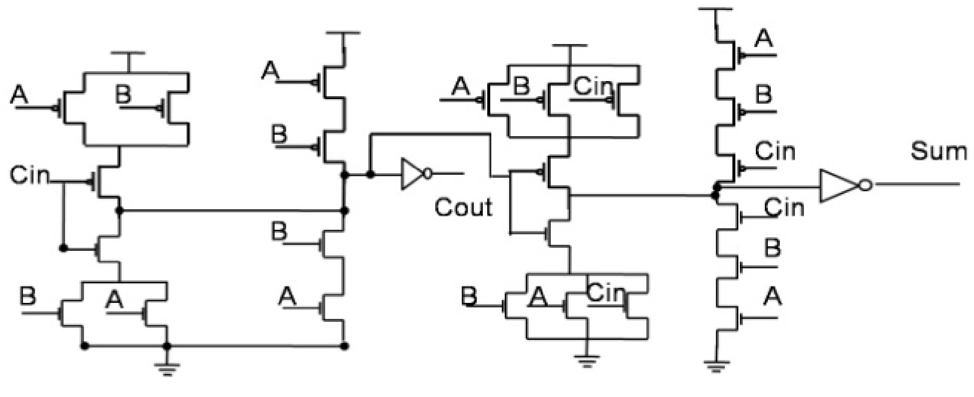


Figure 24: Mirror Adder [10]

The design principle behind the mirror adder is complete symmetry for turn ON/OFF with a maximum of three series transistors for pull UP/DOWN, guaranteeing identical rise and fall transitions. The diffusion capacitances at node $\overline{Cout}$ must be kept at the minimum.

10 GHz Full Adder Design

In the mirror adder, the computation of the SUM needs to wait for the COUT delay.

In order to reduce the delay of the SUM the computation of the SUM and the computation of the COUT can be separated in parallel paths.

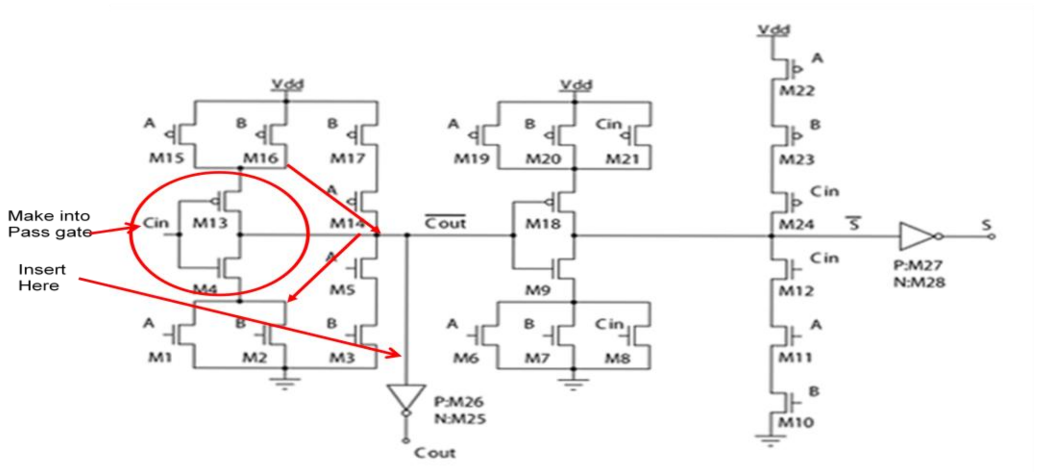


Figure 25: Carry-out bit Computation

As shown in Figure 25, the delay of 2 transistors in series in the COUT computation can be reduced by re-arranging the transistors. M15 and M16 can form a NAND2 gate with M5 and M3. M17 and M14 can form a NOR2 gate with M1 and M2. M13 and M4 become a pass-gate that select to pass either the NAND2 gate output or the NOR2 gate output depending on the 3rd input. The logic computation is shown in Figure 26.

The computation of the SUM consists of computing two inputs with a XNOR2 gates. Then, the output of the XNOR2 gate is inverted. The third input is used to select either the non-inverted output or the inverted output to pass to the inverter. In both the COUT and SUM computation, a pass-gate is used to reduce the number of transistors in series. Then an inverter gate is used to correct the voltage level drop cause by the pass gate. The logic computation for the SUM is shown in Figure 26. The final adder consisting of 27 transistors is shown in Figure 27.

Simulation result from Cadence, for the proposed full adder is shown in Figure 28. The top 3 waveforms are the inputs A, B, and Ci. The bottom 2 waveforms are the outputs SUM and COUT. The simulation matches the truth table shown in Figure 23. The propagation delay for this adder is about 22 ps.

The average power consumption, shown in the simulation in Figure 29, is about 1.77 mW, when operating at 10 GHz.

$$Co = (A \cdot B) + Ci \cdot (A \oplus B)$$
$$Co = A \cdot B + A \cdot B' Ci + A' B \cdot Ci$$
$$Co = A \cdot (B + B' Ci) + A'(B \cdot Ci)$$
$$Co = A \cdot (B + Ci) + A'(B \cdot Ci)$$

$$Sum = (A \oplus B) \oplus C$$
$$Sum = (Ci \oplus B) \oplus A$$
$$Sum = (Ci \oplus B).A' + (Ci \oplus B)' A$$

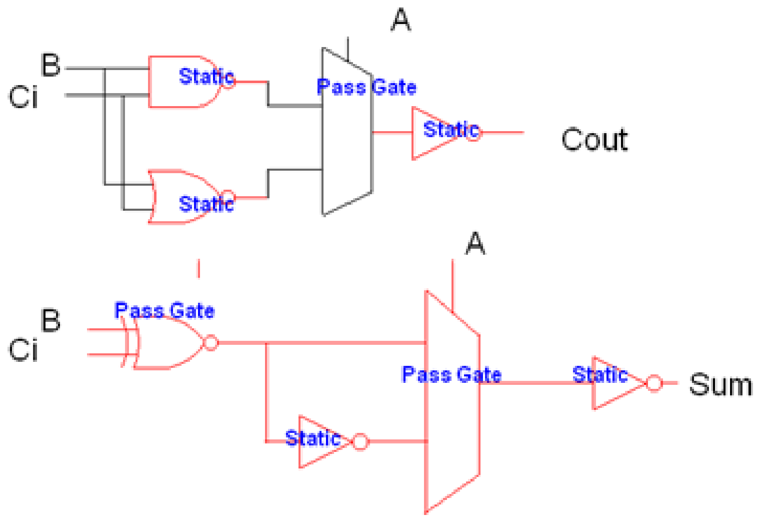


Figure 26: Full adder design

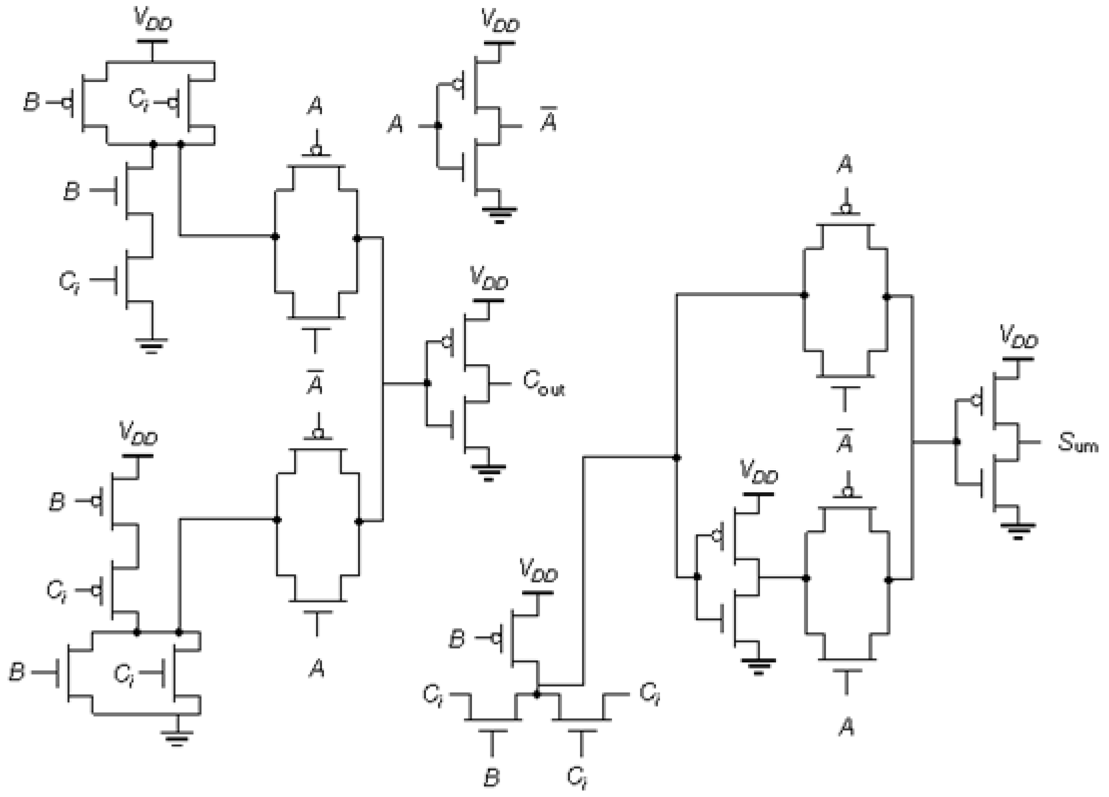


Figure 27: 10 GHz 27T Full-adder Schematic

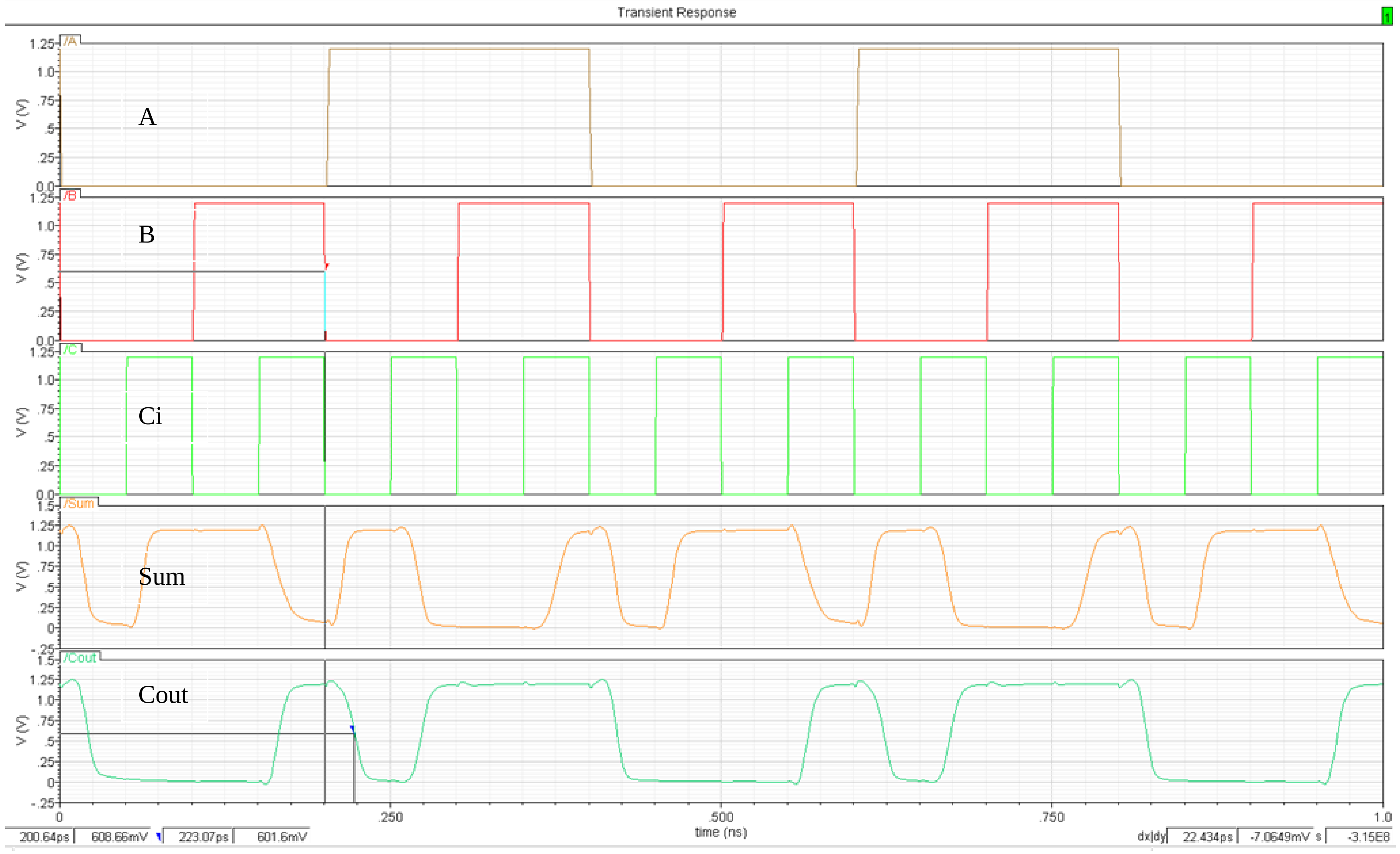


Figure 28: Fast-adder Output waveform

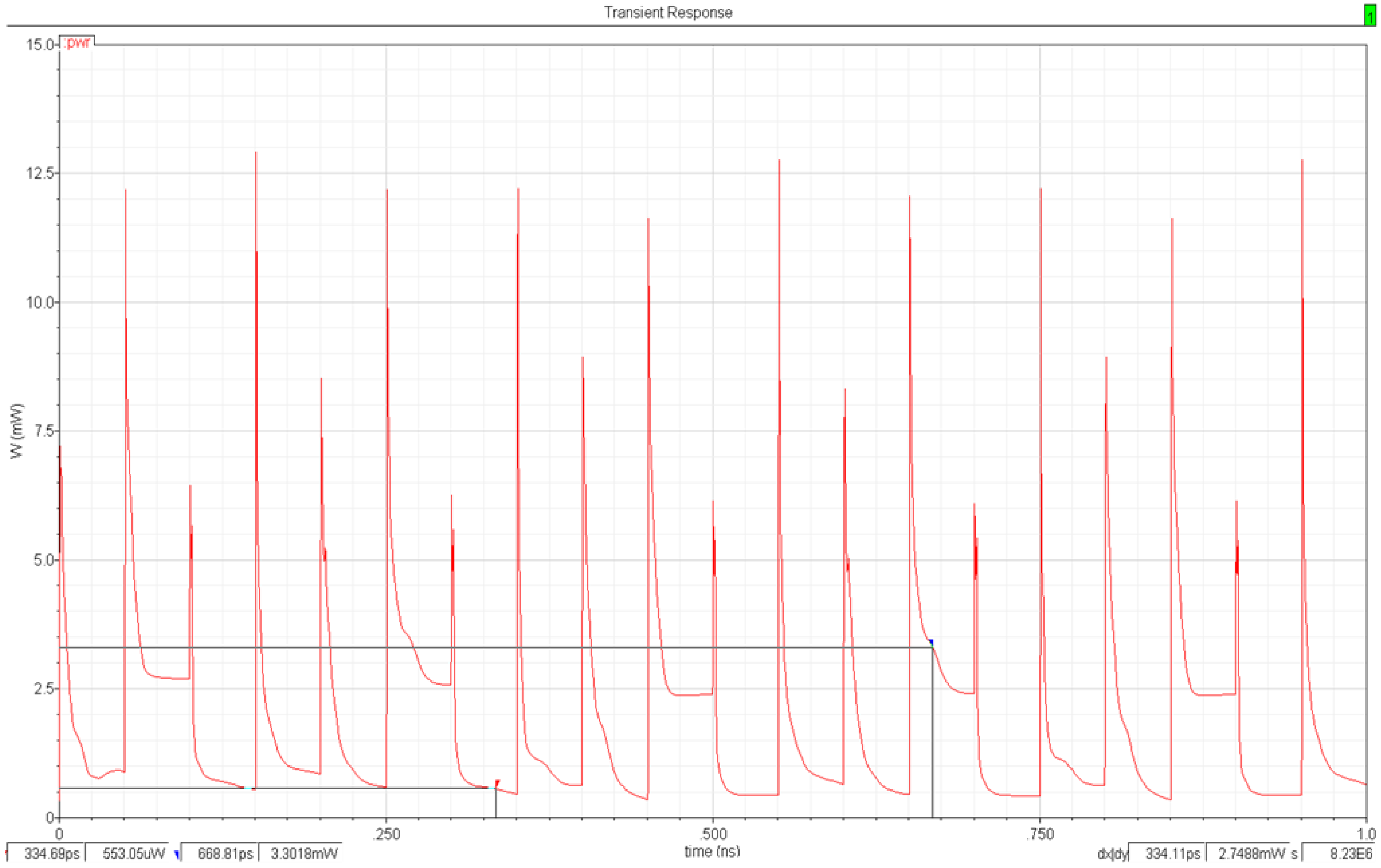


Figure 29: Full Adder Average Power Consumption: 1.77 mW

Phase Accumulator

Essentially, an accumulator is composed of an adder whose sum is loaded into a register. The output of the register is fed back into the adder as shown in Figure 30.

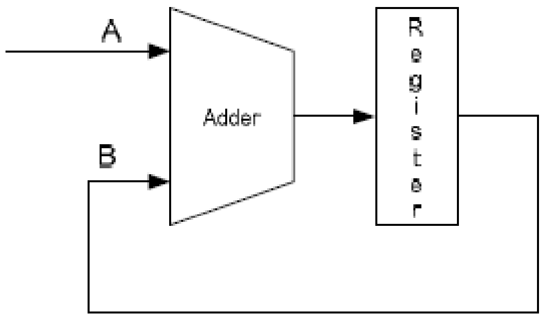


Figure 30: Accumulator Block Diagram

In an n-bit accumulator, "n" number of adders and registers form an n-bit ripple accumulator where the carry-out bit ripples from one accumulator to the next through registers. At every clock tick, the accumulator adds the already accumulated total phase with the frequency tuning word. After a number of clock ticks, and the maximum phase is attained, the accumulator discard the overflow carry-out bit and keep accumulating. The phase accumulator is used as **a** variable increment counter of "n" *bits*. [11] The speed of counting is programmed via the frequency tuning word.

Pipelined Phase Accumulator

The purpose of this section is to highlight the advantages of the pipeline method without which the phase accumulator wouldn't be able to attain higher operating frequency. The illustration is done with the use of an FPGA operating at a much lower frequency. The FPGA evaluation board used has a Xilinx Spartan 3E, and Xilinx ISE software was used to program the FPGA.

The pipelined architecture is an efficient way to reduce the propagation delay in wide arithmetic logic systems, such as adders and accumulators. The concept is to section the full operation into small parts. [13]

Each part is buffered from the next stage by registers which allow all the stages to operate at the same time. One stage just has to wait for the previous stage, rather than all combined delays of the entire chain upfront. The delay is reduced down to one stage regardless how wide the arithmetic logic is. [12]

Figure 31 shows a simulation of 1-bit accumulator in Xilinx. The initial states of all I/Os are zero. Following the transition of *a_in* and *c_in* from 0 to 1, s_out changes state after 7.6 ns. Therefore, the delay of a 1 bit accumulator is about 7.6 ns. Without pipelining, an 8-bit accumulator would have eight times the delay of the 1-bit accumulator.

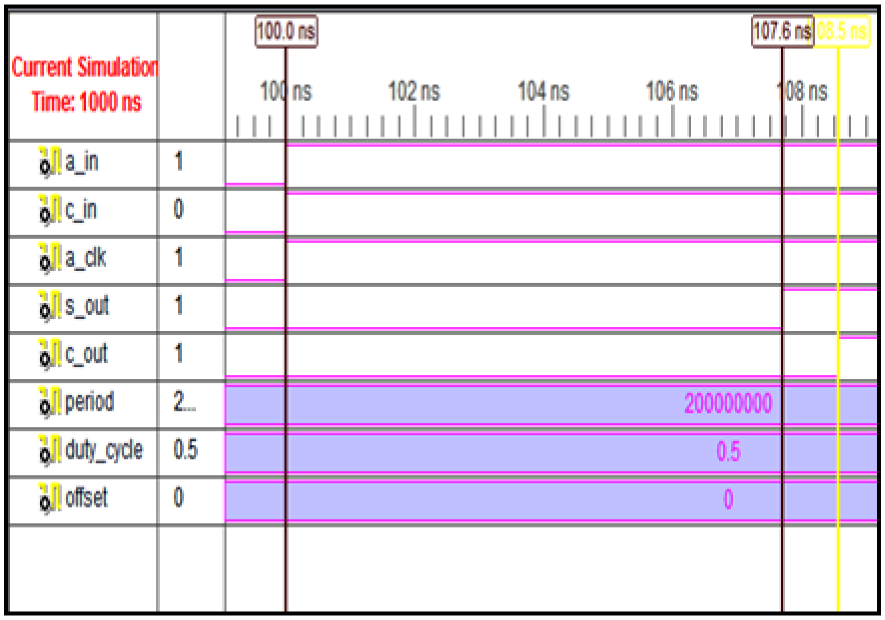


Figure 31: 1 bit accumulator delay, sum = 7.6ns

With pipelining applied to an 8-bit accumulator by introducing registers as shown in Figure 32, the delay of an 8-bit accumulator is measured to be about 7.7 ns in the simulation shown in Figure 34. The overall delay of the pipelined 8-bit accumulator is reduced to the delay of the 1-bit accumulator shown in Figure 34.

Figure 33 shows the outputs of the 8-bit pipelined phase accumulator. With an input tuning word is "*01h*" (decimal=1), the LSB (a_out(0)) changes state at every clock cycle, the LSB-1 (a_out(1)) changes state at every 2 clock cycles, the LSB-2 (a_out(2)) changes state at every 4 clock cycles and so forth…

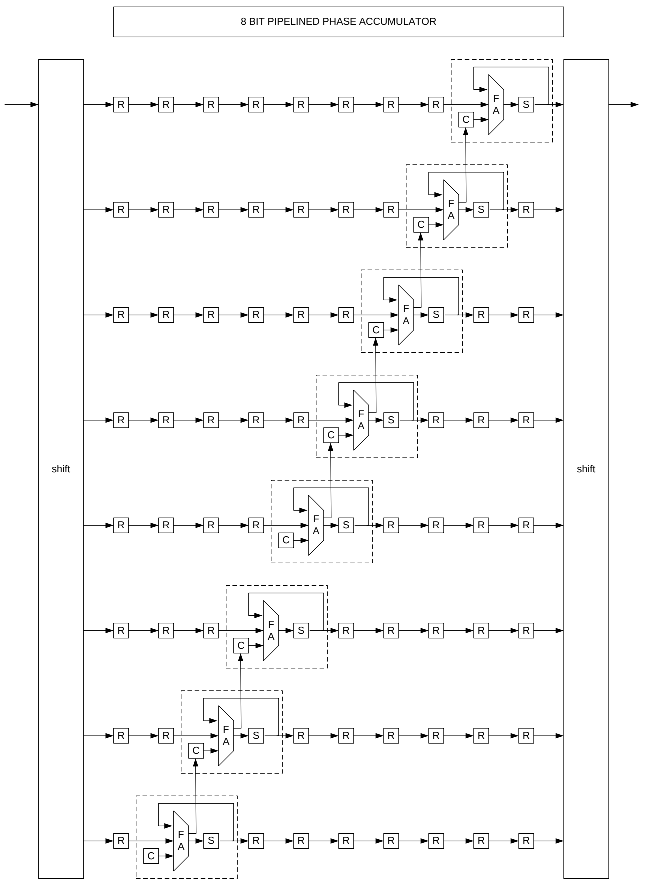


Figure 32: 8 bit Pipelined Phase Accumulator Architecture

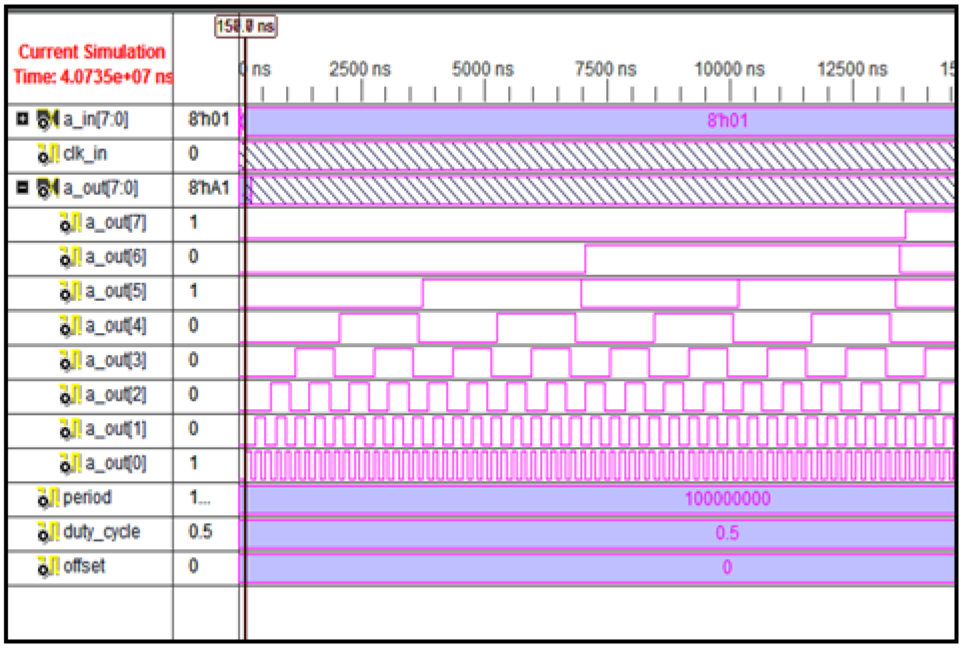


Figure 33: 8 bit accumulator with input word=01h

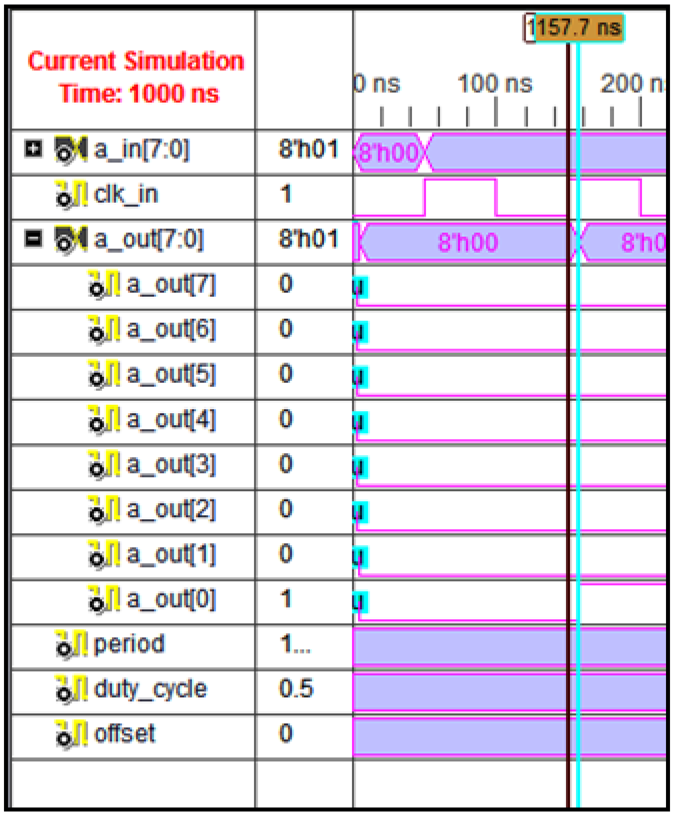


Figure 34: 8 bit accumulator 1st bit delay=7.7 ns

<u>Digital Buffers</u>

In order to drive the inputs of the full-adder, digital buffers are inserted into the feedback loop of the phase accumulator. The digital buffers allow a small size gate to drive a big load with a reduced delay time.

Shown in Figure 35 are two circuits with different delay propagations. The time delay that a small gate takes to drive a large size capacitance C_L could be problematic. The time delay can limit the operating frequency especially in a feedback line. To reduce delay, several buffer stages with different scales of transistors can be inserted into a feedback line as shown on the second circuit in Figure 35.

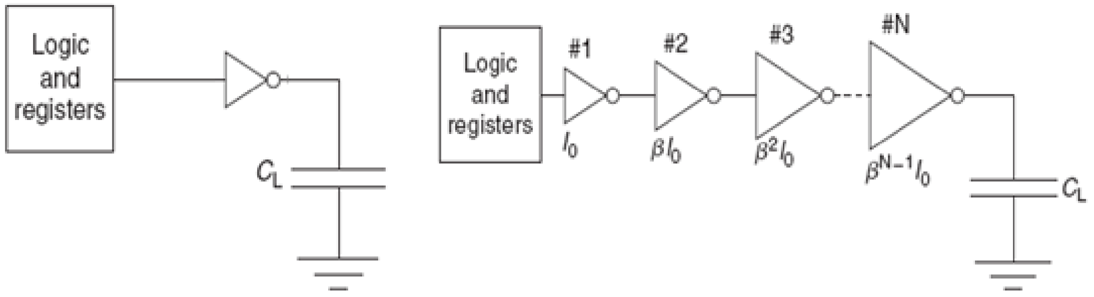


Figure 35: Tapered digital buffers [10]

1-bit Phase Accumulator

The schematic of 1-bit phase accumulator, made up of a full-adder, two registers and several stages of differently scaled digital buffers, is shown in Figure 36.

The 16-bit accumulator is formed from sixteen cells of 1-bit accumulator connected back to back. The schematic of the 16-bit accumulator is shown in Figure 37.

In Figure 38, a chain of sixteen registers is added to the output of the 16-bit accumulator to form the pipeline structure.

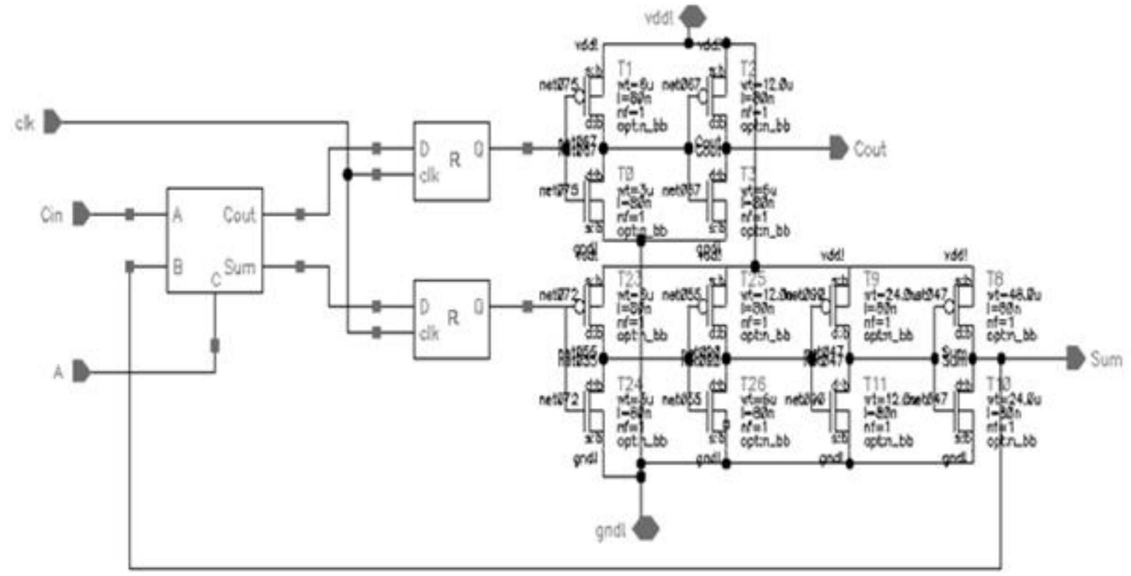


Figure 36: 1-bit accumulator created in Cadence

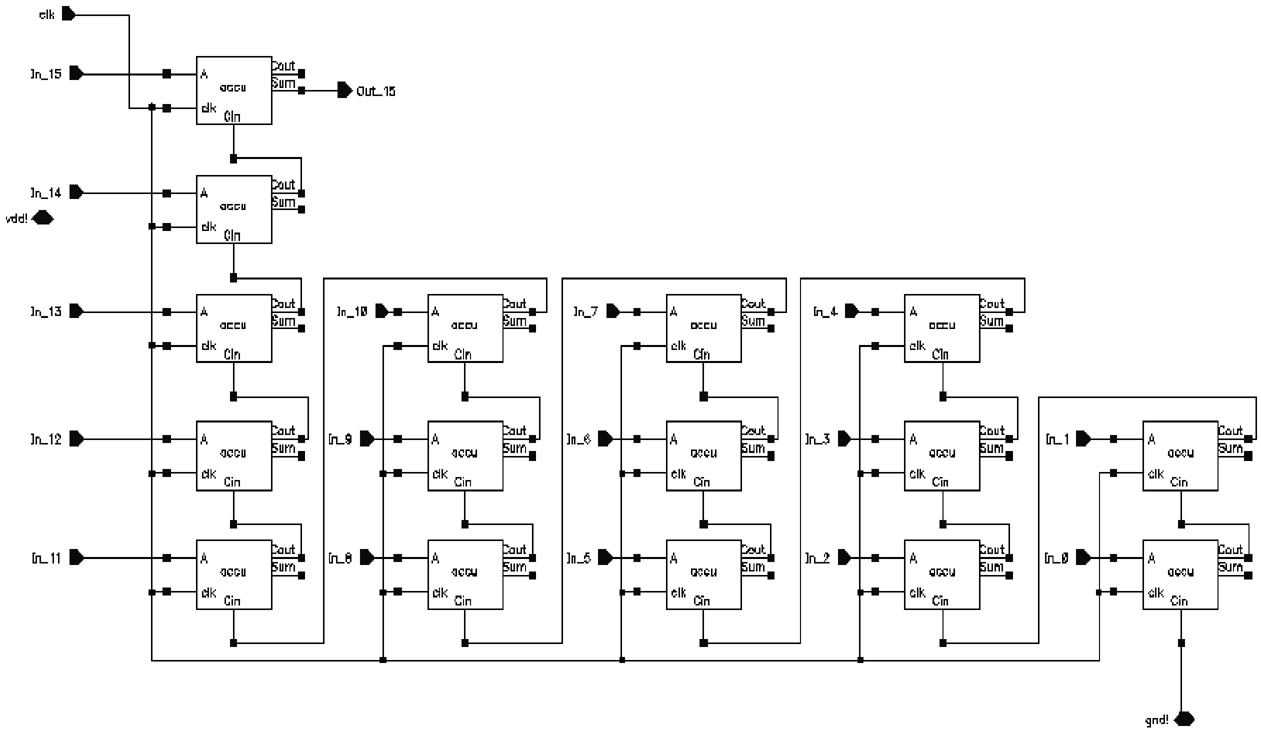


Figure 37: 16-bit Phase Accumulator

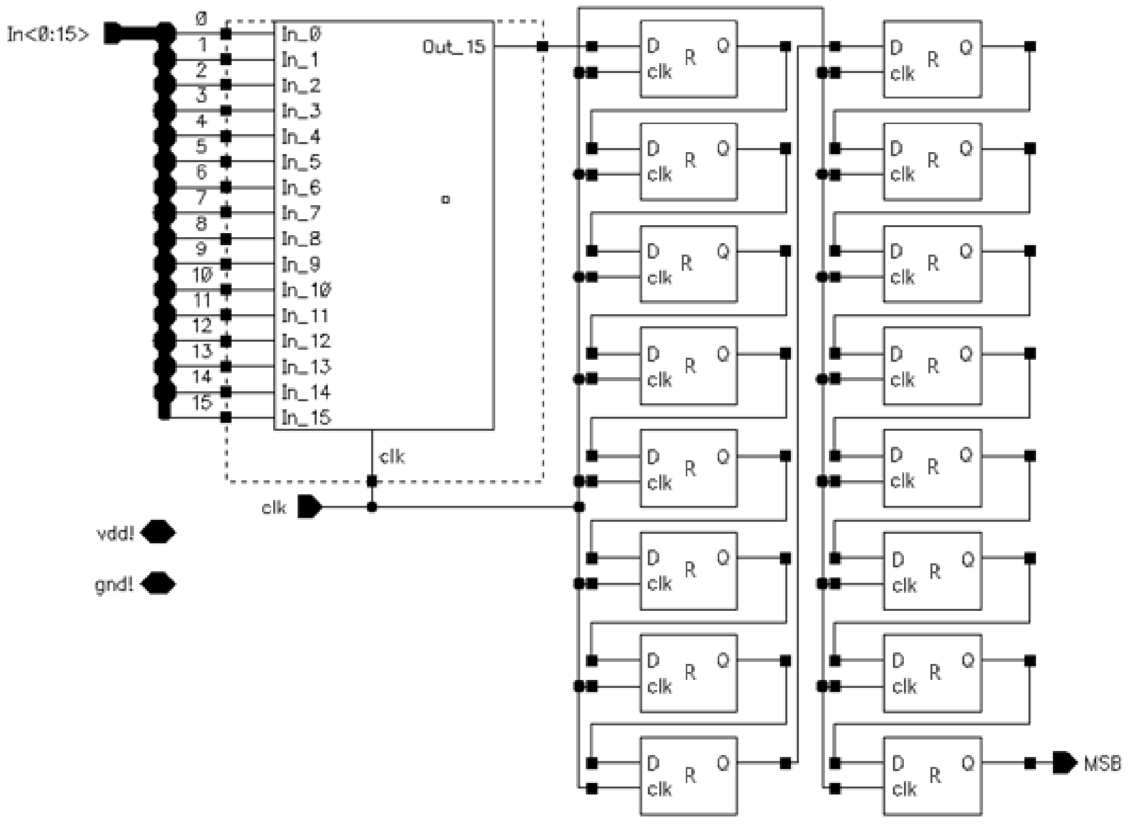


Figure 38: 16-bit Pipelined Phase Accumulator

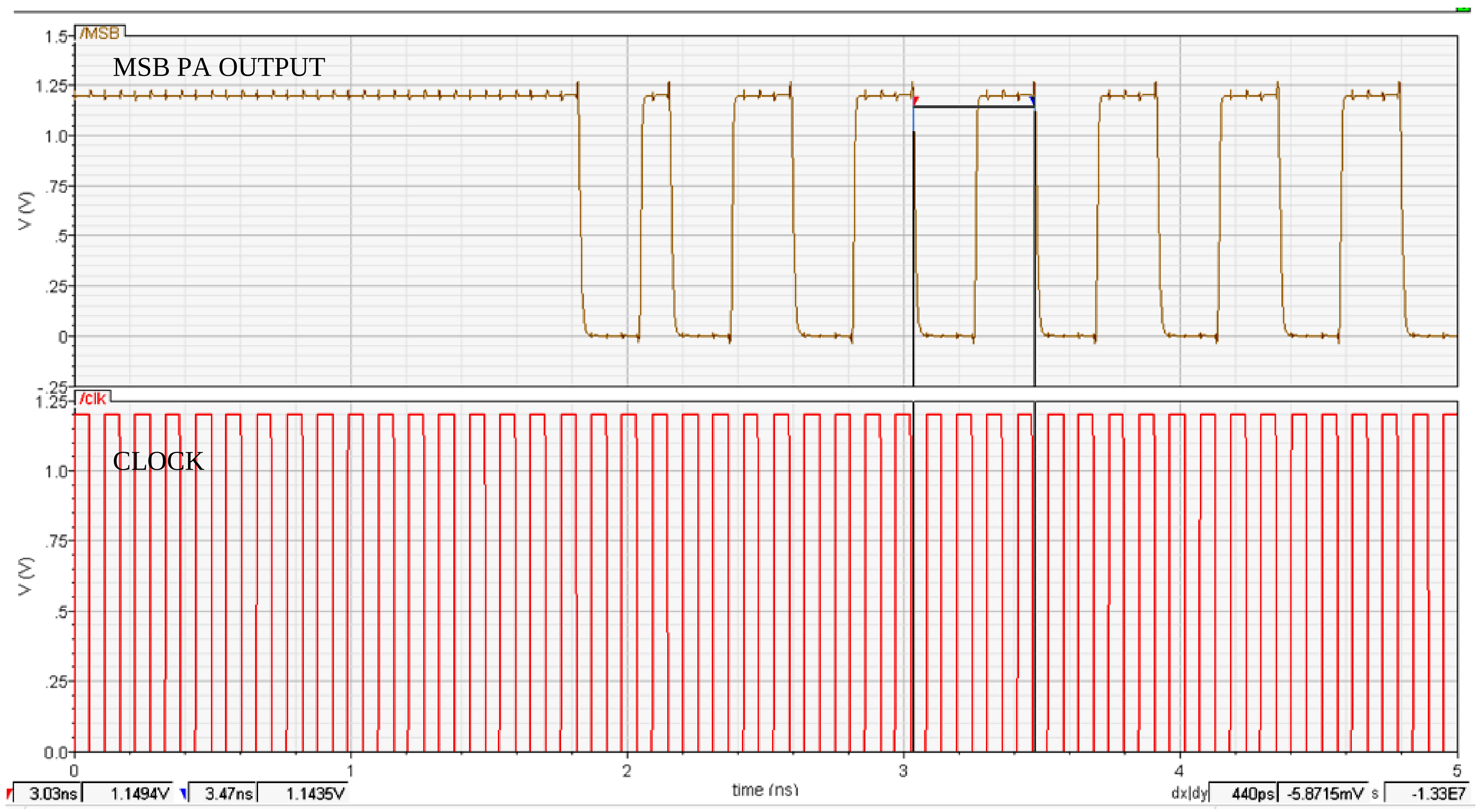


Figure 39: 16-bit PA w/. M=4000h, 1/f=440ps, avg PWR=12mW

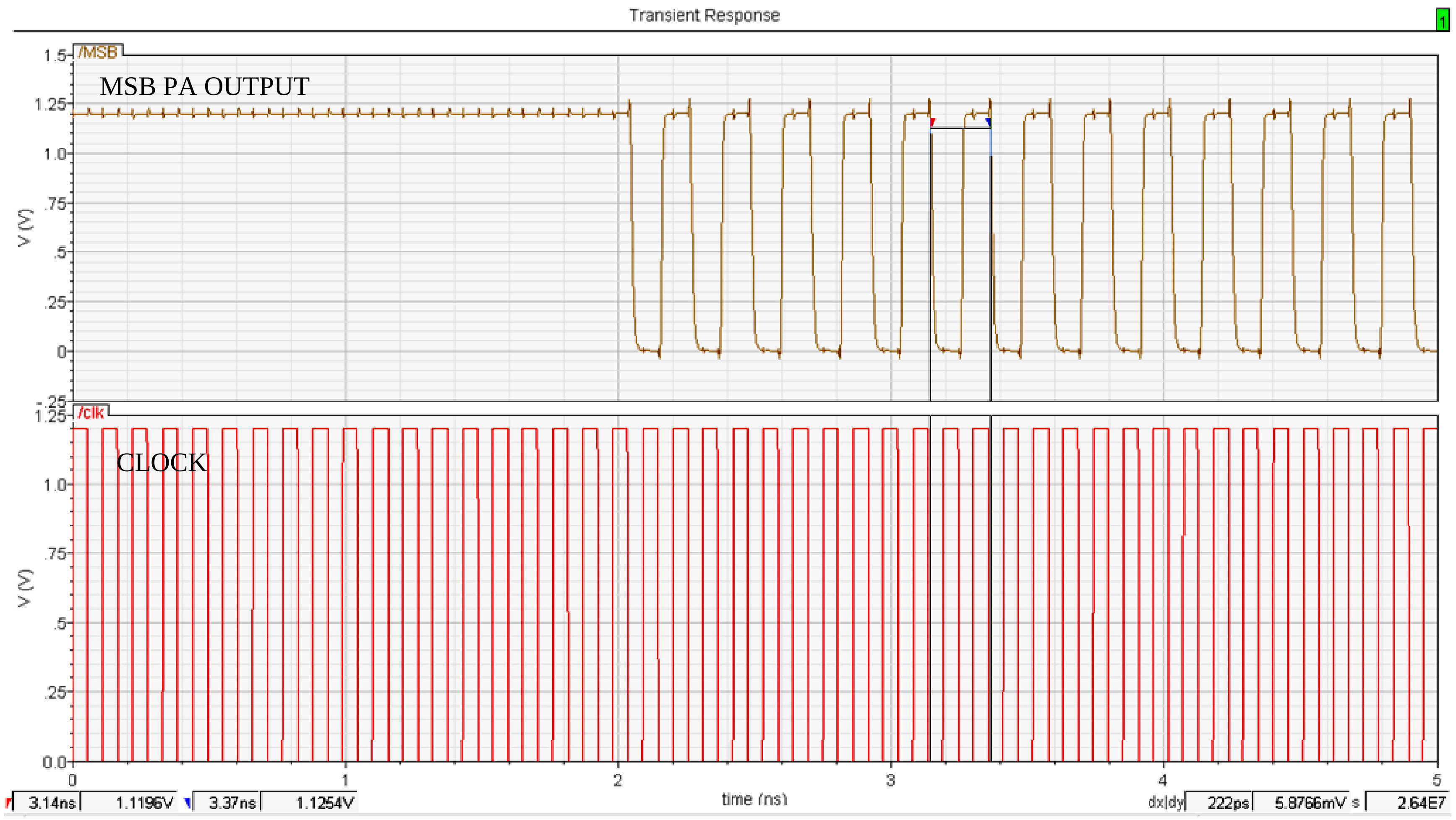


Figure 40: 16-bit PA w/. M=8000h, 1/f=222ps, avg PWR=11mW

Figure 30 and Figure 40 shows the simulated output of the phase accumulator with different input frequency tuning WORDs: 4000h (16384 in decimal) and 8000h (32768 in decimal). A latency of sixteen clock cycles can be observed in both simulations as an effect of the pipeline architecture. The measured periods obtained from the two simulation match the theoretical calculations. The output frequency can be calculated as followed with equation [1-3]:

$$f_{O1} = 9.09\ GHz \times \frac{16384}{2^{16}} = 2.2725\ GHz \qquad [2.1]$$

$$f_{O2} = 9.09\ GHz \times \frac{32768}{2^{16}} = 4.545\ GHz \qquad [2.2]$$

Analog Filter

In this part, the active filter design of the DDS is explained. The filter serves to remove the harmonics frequencies from the square wave output of the phase accumulator in order to convert it into a sine wave. The proposed active filter is a multi-stage staggered-tuning cascode amplifier.

High Frequency Analog IC

Active circuits have replaced the bulky inductors in filters for decades. Gradually, the rise in the operating frequency in electronics has significantly reduced the values of the inductor enough (nano-Henry) to allow the realization of inductors on a chip. [15]

Generally, active filters using operating amplifiers or Operational Trans-conductance Amplifier (OTA) can be used up to 1 GHz. In the gigahertz frequencies, these inductive-less active filters demand semiconductors technologies and process other than CMOS (ex. GaAs, SiGe…).

On the other hand, CMOS analog circuits using on-chip inductors and capacitors can be integrated on the same die as digital circuits and operate at high frequencies. [18]

Resonant circuits and tuned amplifiers which were developed during the vacuum tubes areas for radio communication using large inductors, now reappear in high-frequency analog VLSI applications with on-chip passive components.

Resonant Circuits

A resonant circuit is a circuit that selectively passes a certain frequencies from a source to a load while attenuating all other frequencies. By definition, the resonant circuit is a band-pass filter. The parallel resonance circuits, particularly, are interesting because current sources can be used as an exciter. The basic parallel resonant circuit, or RLC filter, is composed of a capacitor in parallel with an inductor and a resistor.

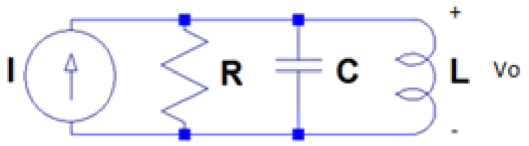


Figure 41: LCR resonator

The admittance of the RLC tank is:

$$Y = G + j\omega C + \frac{1}{j\omega L} = G + j(\omega C - \frac{1}{\omega L}) \qquad [2.3]$$

From the above equation, the resonant frequency can be found as:

$$(C - \frac{1}{L}) = 0 \Rightarrow \omega_0 = \frac{1}{\sqrt{LC}} \qquad [2.4]$$

In the equation [2-3], at the resonant frequency, the admittance is equal to G (or R^{-1}) due to the cancellation of the reactive terms.

A tuned amplifier is a band-pass filter that uses resonant circuits to tune to the desired frequency response. The characteristics of a tuned amplifier can be represented in the frequency domain as shown in Figure 42. How close a tuned amplifier comes to having the characteristics of an *ideal* band-pass circuit depends on the quality (Q) of the circuit.

The quality factor of the resonant circuit is defined as the ratio of the center frequency and the bandwidth. A higher value of Q corresponds to a narrower bandwidth, hence a faster roll-off. The shape factor of a resonant circuit is the ratio of the 60 dB bandwidth and the 3 dB bandwidth. [14]

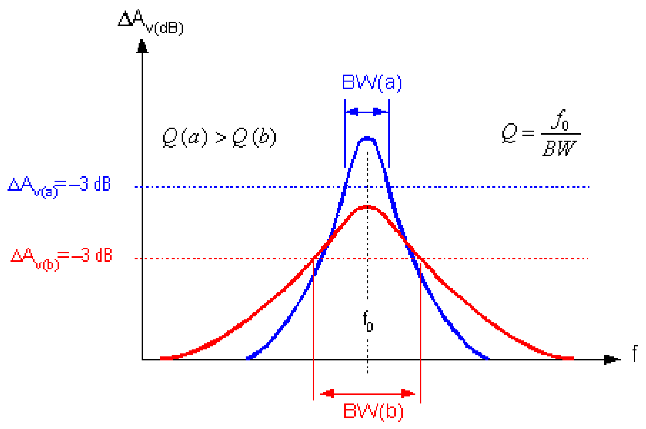


Figure 42: Frequency Response of a Tuned Amplifier [14]

The quality factor can be defined as:

$$Q = \frac{f_0}{BW} = \frac{R}{\sqrt{\left(\frac{L}{C}\right)}}, \quad \text{with } f_0 = \sqrt{f_{C1} f_{C2}} = \frac{1}{2\pi\sqrt{LC}} \qquad [2.5]$$

The RLC filter is a second-order resonator or tank circuit. In consideration of the pole-zero perspective, the impedance of the parallel RLC tank circuit can be re-written in the s-domain as **[** 16]:

$$Z = \frac{1}{Y} = \frac{1}{\frac{1}{R} + sC + \left(\frac{1}{sL}\right)} = \frac{1}{C}\frac{s}{s^2 + s\left(\frac{1}{RC}\right) + \frac{1}{LC}} = \frac{1}{C}\frac{s}{(s - s_{p1})(s - s_{p2})} \qquad [2.6]$$

where,

$$s = -\eta \pm \sqrt{\eta^2 - \omega_0^2} \triangleq -\frac{1}{2RC} \pm \sqrt{\left(\frac{1}{2RC}\right)^2 - \frac{1}{LC}} \qquad [2.7]$$

High Frequency Amplifiers

MOSFET at High Frequency

Operation at high frequency requires the examination of the parasitic elements of the active device. The parasitic capacitances of a MOSFET cause the gain to fall at high frequency. The gate capacitive effect, which is mainly the cause for reducing the MOSFET performance, is composed of three parasitic capacitances C_{gs}, C_{gd}, and C_{gb}. Table 3 shows the calculation to obtain the capacitance values from the geometry of the MOSFET.

Table 3: Approx. of intrinsic MOS gate capacitance [10]

	Triode	**Saturation**	**Cutoff**
C_{gs}	$\frac{1}{2}WLC_{OX}$	$\frac{2}{3}WLC_{OX}$	0
C_{gd}	$\frac{1}{2}WLC_{OX}$	0	0
C_{gb}	0	0	WLC_{OX}

The junction capacitances, which are composed of C_{sb} and C_{db}, also contribute to high frequency gain degradation but at a lower degree.[15] To simplify calculation, only C_{gs} and C_{gd} are includes in the high frequency small signal model of the MOSFET. Figure 43 shows C_{gs} and C_{gd} in the small signal model of a MOSFET operating at high frequency.

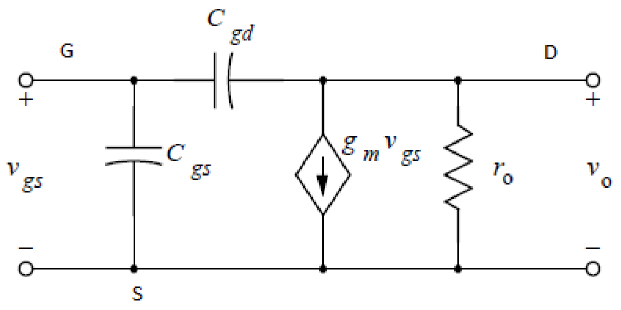


Figure 43: NMOS High Freq. Small-signal Model [15]

Tuned Amplifiers

Tuned amplifiers achieve high frequency operation by operating within a narrow bandwidth. Additionally, the tuned filters help to attenuate spurious noise outside the operating. The simplest tuned amplifier, shown in Figure 44, consists of a MOSFET and a parallel RLC resonant circuit as the load.

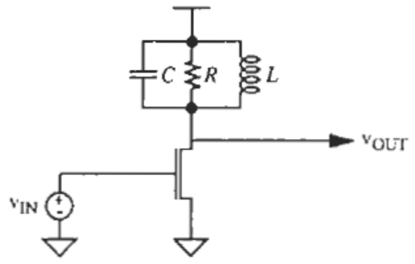


Figure 44: Simple Tuned Amplifier [19]

At low frequency, the load inductor acts as a short, causing the gain to be zero. At high frequency, the load capacitor acts as a short and the gain also goes to zero. At the resonant frequency the gain is $g_m R$ with the cancellation of the reactance terms. The bandwidth is equal to $\frac{1}{RC}$.

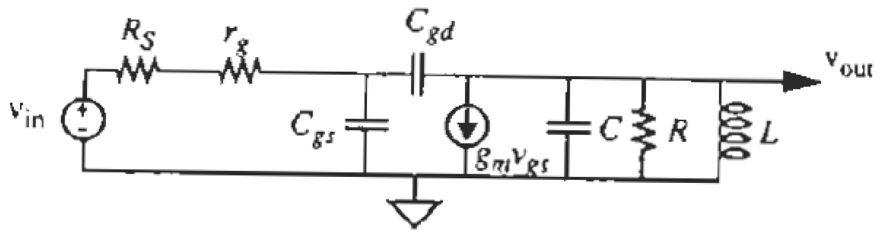


Figure 45: Simple Tuned Amp. Small-signal Model [19]

The small signal model of the simple tuned amplifier, shown in Figure 45, helps to determine the cutoff high-frequencies. The internal equivalent capacitance C_{eq} of the MOSFET can be quite large due to the Miller capacitance multiplication effect. This parasitic capacitance can shift downward the resonant frequency of the output RLC tank circuit. The equation for this C_{eq} is [19]:

$$C_{eq} = C_{gd}\left[1 + g_m R_{eq}\right] = C_{gd}\left[1 + g_m\left(R_S + r_g\right)\right] \qquad [2.8]$$

<u>Cascode Configuration</u>

Cascading two transistors on top of each other, in a cascode configuration helps to reduce the Miller effect. The cascode amplifier is a combination of a common-source configuration with a common-gate configuration. De-tuning effect cause by the Miller capacitance is also eliminated; making the cascade configuration an ideal for use at high frequencies. Figure 46 shows the circuit of the tuned cascode amplifier. The small signal model is shown in Figure 47.

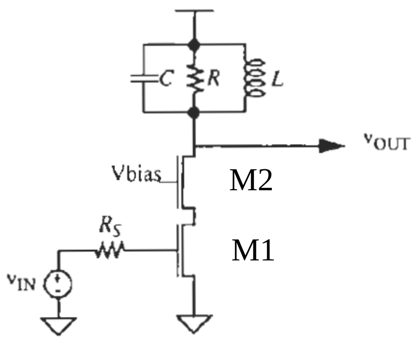


Figure 46: Tuned Cascode Amplifier [19]

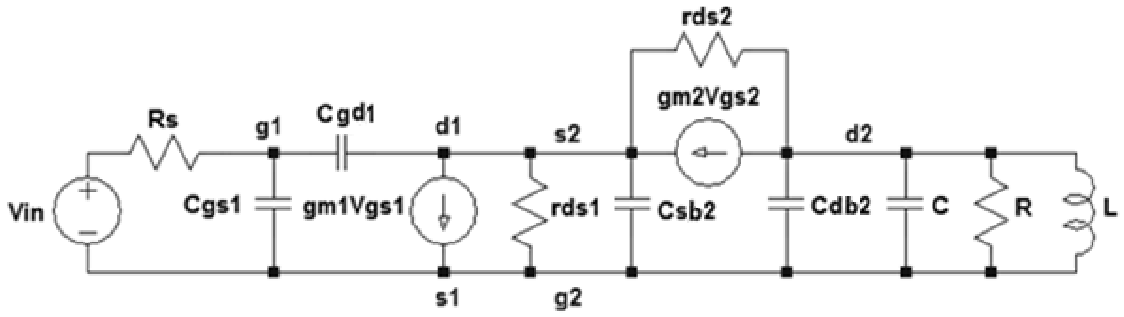


Figure 47: Tuned Cascode Amplifier Small-signal Model

Staggered Configuration

In consideration of the bandwidth performance, a single tuned amplifier may not satisfy the frequency response requirement. The bandwidth can be improved by using multi-stage filters in a stagger-tuning manner. The resulting frequency response offer maximal flatness around the center frequency, a wider bandwidth, and a faster roll-off in the cut-off band.

The process of designing a stagger-tuning amplifier starts with building a stagger-tuning band-pass Butterworth filter with “n” number of poles corresponding to “n” number of stages. The Butterworth filter offers the maximum flatness in the pass-band frequencies. [14]

If needed, the second step consists of transforming the Butterworth filter into a Chebychev filter to obtain stepper roll-off in the cutoff band, but, in exchange for some ripples in the band-pass.

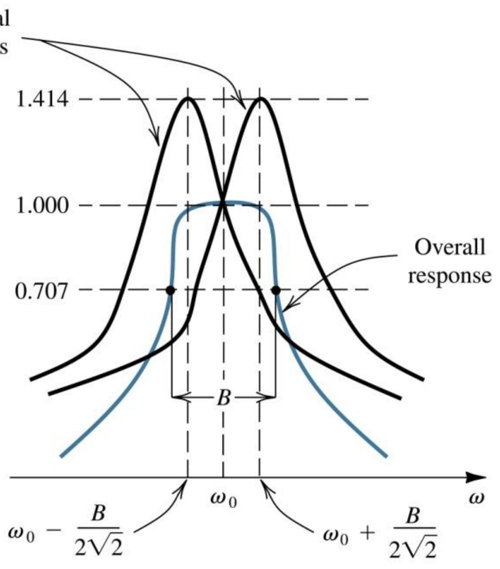


Figure 48: Stagger Tuning [14]

Step 1: Butterworth

The Butterworth filter contains only poles. In the pole-zero graph, the poles of a Butterworth filter are located on a circle with radius ω_0 and they are spaced apart by an angle of $\frac{180^0}{n}$, with "n" being the order of the filter (number of poles). The first pole is located $\frac{180^0}{2n}$ from the jω axis, as shown in the Figure 49 below.

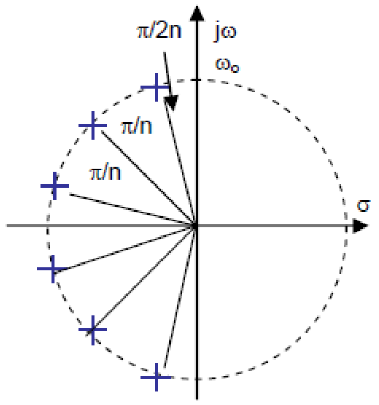


Figure 49: Butterworth Filter with six poles [14]

Figure 49 shows the location of the poles of a normalized Butterworth on the pole-zero graph for filters that are of 1st order to 8th order.

Table 4: Poles of the Normalized Butterworth Polynomials

Order	**Poles**
1	$-1 \pm j\ 0$
2	$-0.707 \pm j\ 0.707$
3	$-1 \pm j\ 0$, $-0.5 \pm j\ 0.866$
4	$-0.924 \pm j\ 0.383$, $-0.383 \pm j\ 0.924$
5	$-1 \pm j\ 0$, $-0.809 \pm j\ 0.588$, $-0.309 \pm j\ 0.951$
6	$-0.966 \pm j\ 0.259$, $-0.707 \pm j\ 0.707$, $-0.259 \pm j\ 0.966$

For example, a design consisting of a three staggered-tuning stages Butterworth filter is needed. The center frequency of this filter is 1 GHz and the bandwidth is 200 MHz.

One of the three stages is tuned at the center frequency. The tuning frequencies of the other two stages are staggered one at a lower frequency and one at a higher frequency. Their equal distances away from the tuned frequency can be calculated:

$$\frac{BW}{2} \times cos\left(\frac{\pi}{6}\right) = 100\ Mhz \times 0.866 = 86.6Mhz \qquad [2.9]$$

Step 2: Butterworth to Chebyshev Transformation

The Chebyshev filter, as shown in Figure 50, is very similar to the Butterworth filter, except that the poles of a Chebyshev filter lie on an elliptical shape instead of a circle. There is a simplified method to transform the poles of a Butterworth filter directly into the poles of a Chebyshev filter, while keeping the same bandwidth. For an "n" order Chebyshev filter with "r" amount ripple (in dB), the real magnitude of the Chebyshev pole ($|P_C|$) can be calculated from the real magnitude of the Butterworth pole ($|P_B|$) [15]:

$$|P_C| = |P_B|\, tanh\, a\, , \text{with } a = \frac{1}{n} sinh^{-1} \frac{1}{\sqrt{10^{(\frac{r}{10})} - 1}} \qquad [2.10]$$

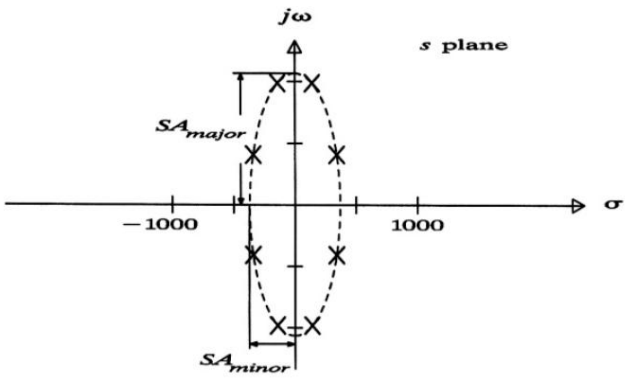


Figure 50: Chebyshev poles location [15]

```
%Coefficient of conversion Butterworth-Chebyshev
r=[.05 0.1 0.2 0.3 0.4 0.5];                     %ripple(dB)
n=[2 3 4 5 6 7 8];                               %# of stages
for m=1:6
    for k=1:7
        epsilon(m,k)=sqrt(10^(r(m)/10)-1);
        a(m,k)=(1/n(k))*asinh(1/epsilon(m,k));
        x(m,k)=tanh(a(m,k));
    end
end
```

Figure 51: Matlab Code, Butt. To Chev. Conv. Coeff.

In Figure 51, a MATLAB code is created to compute the "*tanh a*" coefficient for different order filter and different level of ripple in the pass-band. The result is shown in Table 5.

For example, for a design requiring a three stages filter with a maximum pass-band ripple of 0.5dB, the conversion coefficient is 0.53089.

Table 5: Butterworth to Chebychev Conversion Coeff, tanh a

r(dB)	n=2	n=3	n=4	n=5	n=6	n=7
0.05	0.89817	0.75094	0.62387	0.52633	0.45227	0.39516
0.1	0.85896	0.69604	0.56808	0.47441	0.40514	0.35258
0.2	0.8062	0.63159	0.50652	0.41896	0.35576	0.30849
0.3	0.76771	0.58923	0.46789	0.38498	0.32591	0.28206
0.4	0.73652	0.55705	0.43934	0.36021	0.30432	0.26305
0.5	0.70994	0.53089	0.41657	0.34065	0.28736	0.24816

Filter Design

Three Stages Stagger-Tuning

The following calculations were intended for the design of a three stage stagger tuning filter with the following specifications:

EXAMPLE:

- Center frequency: 4.545 GHz
- Ripple: 0.5 dB
- Bandwidth: 400 MHz
- 1 nF fixed inductor

$$\Delta f = \frac{BW}{2} = \frac{400\ MHz}{2} = 200MHz$$

[2.11]

Stage 1:

$$f_1 = f_0 - \Delta f cos\left(\frac{\pi}{6}\right) = 4.545\ GHz - (200\ MHz \times 0.866025) = 4.37GHz$$

[2.12]

$$Q_1 = \frac{f_1}{BWsin(30^o)} \times \frac{1}{tanh\ a} = \frac{4.37GHz}{400\ MHz \times 0.5} \times \frac{1}{0.53089} = 41.2$$

[2.13]

$$R_{p1} = L_1 \omega_1 Q_1 = 1\ nH \times 2\pi(4.37\ GHz) \times 41.2 = 1131.24\ \Omega$$

[2.14]

$$C_1 = \frac{1}{L \times (2\pi f_1)^2} = \frac{1}{1\,nH \times (2\pi \cdot 4.37\ GHz)^2} = 1.325pF$$

[2.15]

<u>Stage 2:</u>

$$f_2 = f_0 - \Delta f cos\left(\frac{\pi}{2}\right) = 4.545\ GHz - (200\ MHz \times 0) = 4.545GHz$$

[2.16]

$$Q_2 = \frac{f_2}{BWsin(90^o)} \times \frac{1}{tanh\,a} = \frac{4.545GHz}{400\ MHz \times 1} \times \frac{1}{0.53089} = 21.4$$

[2.17]

$$R_{p2} = L_2\omega_2 Q_2 = 1\,nH \times 2\pi(4.545\ GHz) \times 21.4 = 611.32\ \Omega$$

[2.18]

$$C_1 = \frac{1}{L \times (2\pi f_2)^2} = \frac{1}{1\,nH \times (2\pi \cdot 4.545\ GHz)^2} = 1.226pF$$

[2.19]

<u>Stage 3:</u>

$$f_3 = f_0 - \Delta f cos\left(\frac{5\pi}{6}\right) = 4.545\ GHz + (200\ MHz \times 0.866025) = 4.72GHz$$

[2.20]

$$Q_3 = \frac{f_3}{BWsin(30^o)} \times \frac{1}{tanh\,a} = \frac{4.72GHz}{400\ MHz \times 0.5} \times \frac{1}{0.53089} = 44.4$$

[2.21]

$$R_{p3} = L_3\omega_3 Q_3 = 1\,nH \times 2\pi(4.72\ GHz) \times 44.4 = 1317.59\ \Omega$$

[2.22]

$$C_1 = \frac{1}{L \times (2\pi f_3)^2} = \frac{1}{1\,nH \times (2\pi \cdot 4.72\ GHz)^2} = 1.138pF$$

[2.23]

Single Stage Filter Schematics

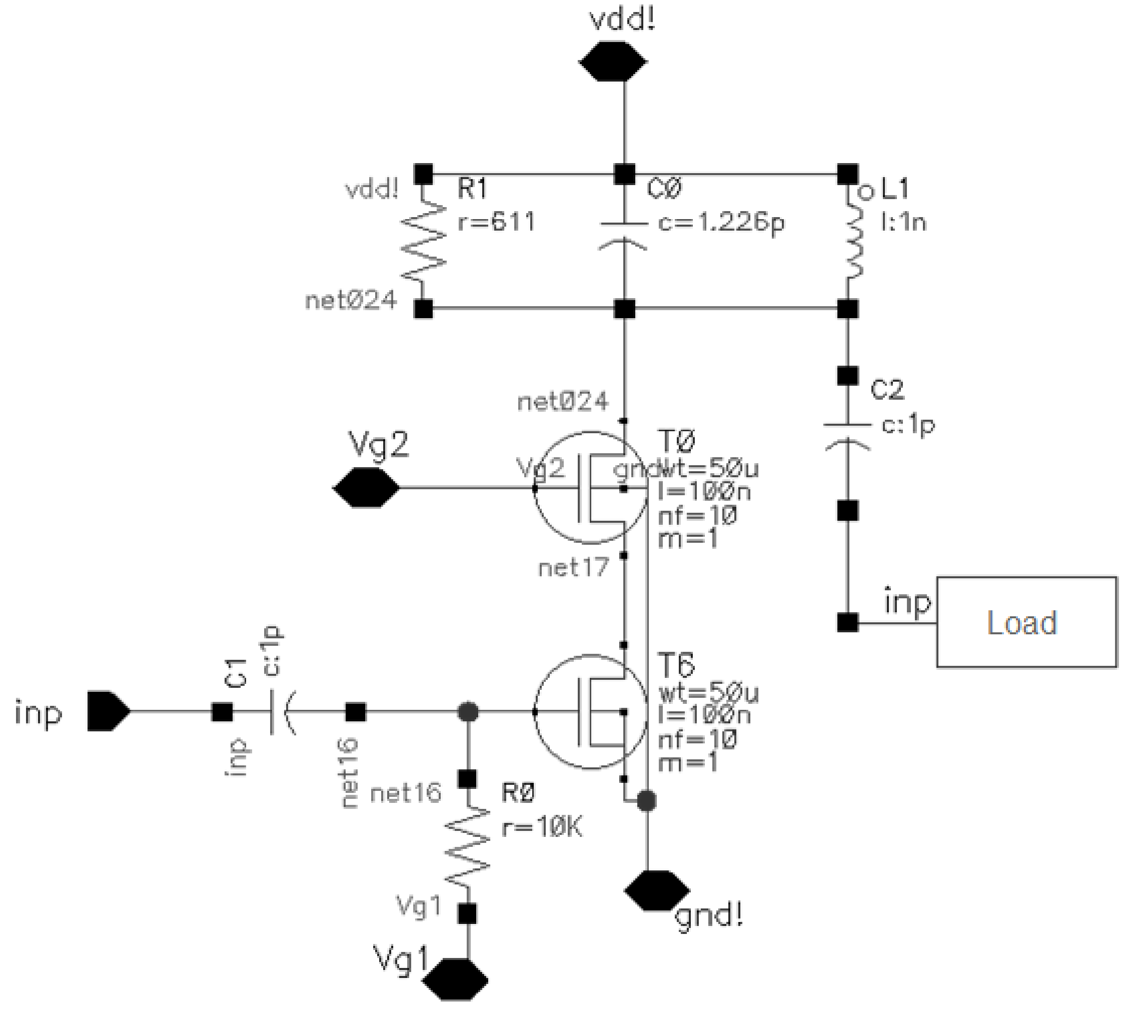


Figure 52: Single Stage Filter tuned for 4.545 GHz

The schematic of a one stage tuned filter, drawn with Cadence, is shown in Figure 52. This filter is tuned to 4.545 GHz with the tank circuit consisting of L1, C0, and R1. R1 is set to 611 Ω, C0 is set to 1.226 pF, and L1 is set to 1 nH, approximately. C1 and C2 are 1pF capacitor used for DC voltage decoupling. Vg1, Vg2 and R0 bias ON the N-channel MOSFETs, T0 and T6. Vg1 is set to 0.45 Vdc and Vg2 is set to 0.95 Vdc.

Transistors, T0 and T6, are set in the cascode configuration. Their widths are set to 50 µm.

Figure 53 shows the frequency response of this one stage active filter.

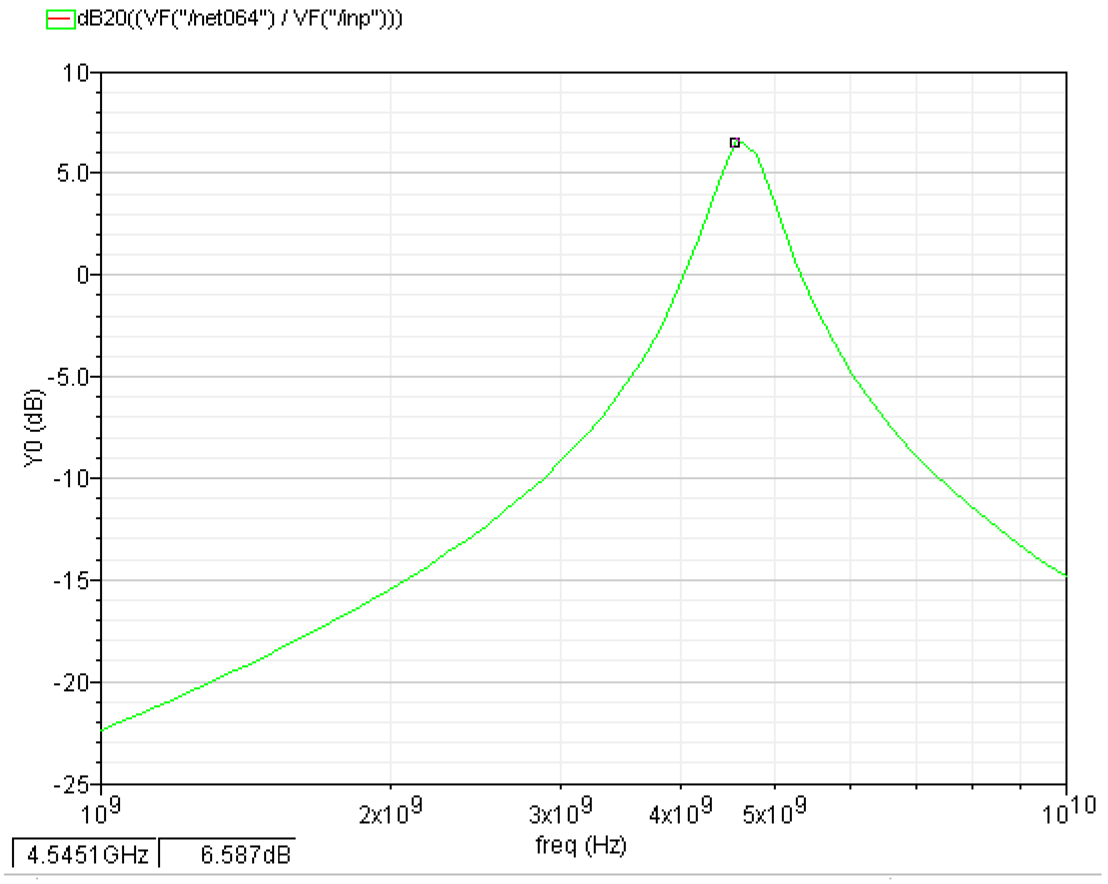


Figure 53: Single Stage Tuned Filter Frequency Response

<u>Three Stages Stagger-Tuning</u>

The three stages staggered-tuning cascode amplifier, shown in Figure 55, consists of three single tuned amplifiers connected back to back, in series. The values of the passive components of the resonant circuits were selected based on calculations done from equations [2.11] to [2.23]. 1 pF capacitors are used for DC voltage decoupling.

Figure 54 shows the frequency response of each of the stage filter (black, red, purple).

The overall response (green), shown in Figure 54, is the sum of the three frequency response of the individual stages.

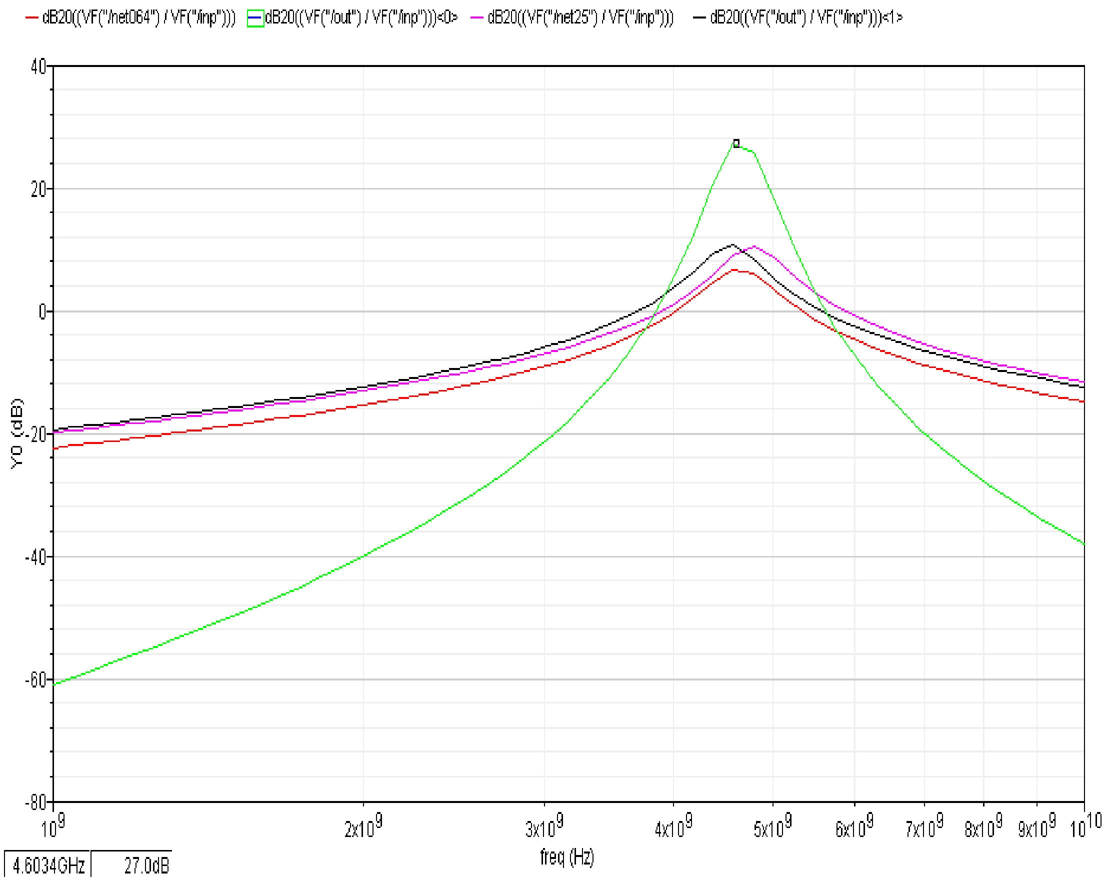


Figure 54: Freq. response of the 3-stages filter

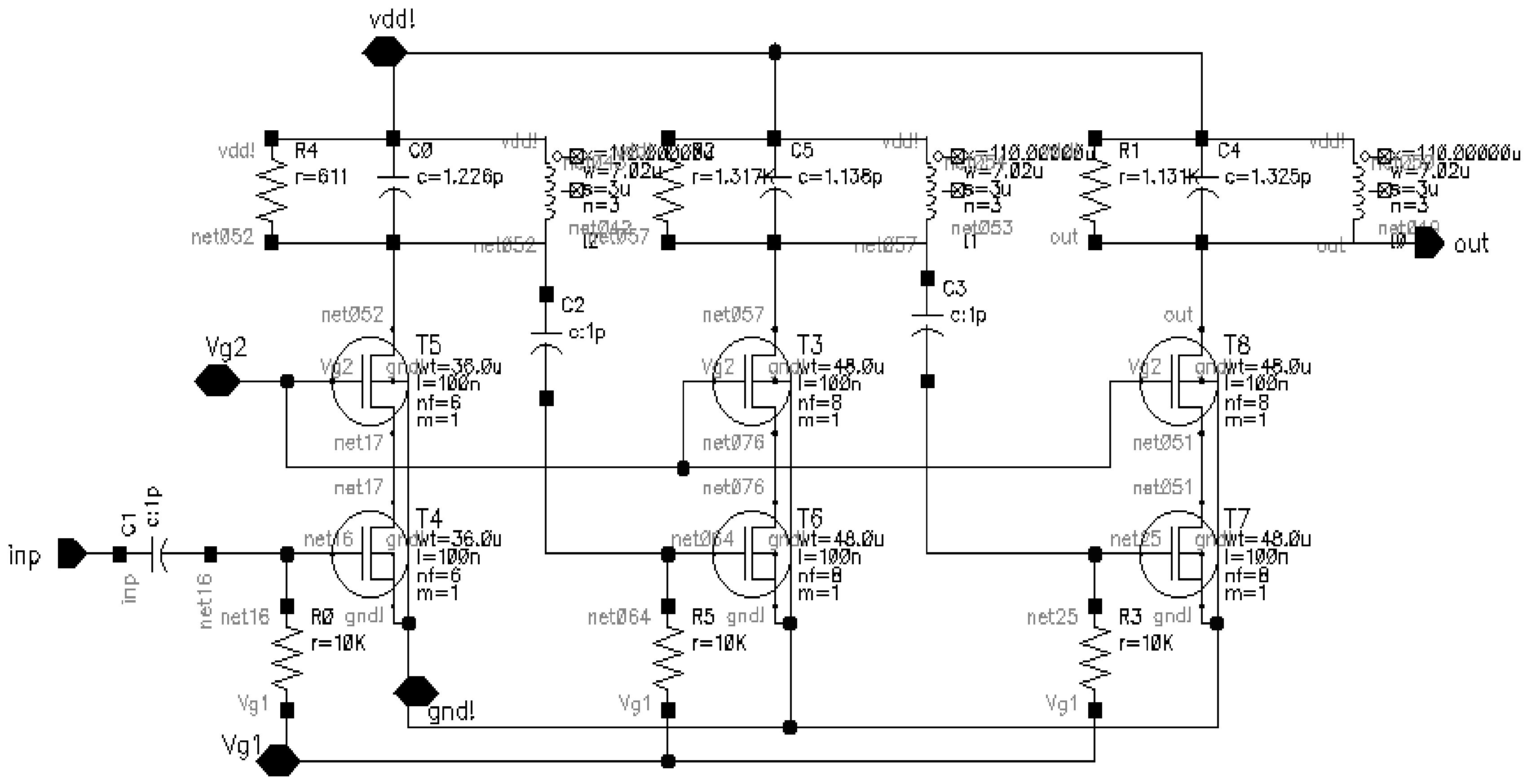


Figure 55: 3 Stages Stagg.-tuning filter (4.545 GHz)

Output Driver

Power Amplifier

The purpose of the power amplifier (PA) is to give the DDS the capability of driving an output load impedance of 50 Ω. The alternative method to using power amplifiers is to use analog buffers consisting of operational amplifiers. The design of such operational amplifier operating at high-frequency is a challenging task, but, it can achieve wider frequency response.

Power amplifiers can be classified into two categories based on how the active devices behave. The first category is the trans-conductance PA, which acts as a voltage-controlled current source. The output current of trans-conductance PAs is controlled by the input voltage.

The second category of power amplifiers is the switching PA (Pulse Width Modulation) which uses the active device as a switch to modulate the output voltage or current. [18]

Classes A, AB, B, and C belong to the trans-conductance PA family. Classes D and E are switching PAs. Class F can be either switching PA, or trans-conductance PA, depending on how hard the active devices are driven.

Within the trans-conductance amplifier family, many distinctions in the operation of the amplifier separate the amplifiers into different classes. In the class A amplifiers, the transistor is biased ON 100% of the times. The transistor has a conduction angle of 360° (100% of a cycle). Typical waveforms of a class A amplifier are shown in Figure 56. Since the device is turned ON 100% of the time, no nonlinearities due to signal clipping are introduced. For this reason, the class A amplifier is often considered the most linear of all of the class A amplifiers. However, the efficiency of the class A PA is very poor and rarely exceeds 30-40%.

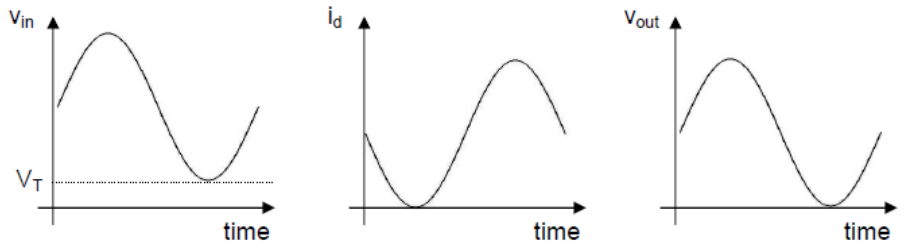


Figure 56: Class A PA waveforms [19]

Class A Amplifier Design

The design requirement for the amplifier is to be able to drive a 50 Ω load with 10 dBm minimum output power with a supply voltage of 1.2 Vdc. The following calculation determines if the supply voltage can deliver sufficient power to the load without impedance transformation:

$$P_{max} = \frac{V_{DD}^2}{2R_L} = \frac{1.2^2}{2 \times 50} = 0.0144\, W$$

(with full output swing) [2.24]

$$dBm = 10\, log_{10}\left(\frac{P_{max}}{1\, mW}\right) = 11.58\, dBm$$

[2.25]

From the obtained results, the amplifier isn't required to have any impedance transformation.

With a load of 50 Ω, the peak current is:

$$\frac{V_{DD}}{R} = \frac{1.2}{50} = 0.024\, A$$

[2.26]

The PA efficiency is predicted to be:

$$\eta = \frac{P_o}{P_{DC}} = \frac{.0144\, W}{.024\, A \times 1.2V} = 50\%$$

[2.27]

The active device is predicted to consume the following power:

$$P_{LOSS} = P_{DC} \times (1 - \eta) = 0.0144\, W$$ [2.28]

Figure 57 shows the schematic of the class A amplifier drawn with Cadence. The amplifier consists of two DC decoupling capacitors, C1 and C2, a resistor, R5, used to bias the MOSFET ON with 0.95 Vdc at Vg1, a choke inductor, set at approximately 25 nH, and a N-channel MOSFET with the width set to 42 µm.

Figure 58 shows the frequency reponse of the PA, the amplifier exhibit relatively flat frequency response from 1

GHz to 10 GHz. The lower 3dB frequency of the PA is approxmimatley about 314MHz.

Figure 59 shows the transient response and the power dissipation of the PA, when operating at 4.545 GHz. The output is 180° out of phase with the input. The output has approximaltely a voltage swing of $\pm$ 500 mV. The power of the PA peaks at 14.7 mW and the average power dissipation is about 13 mW.

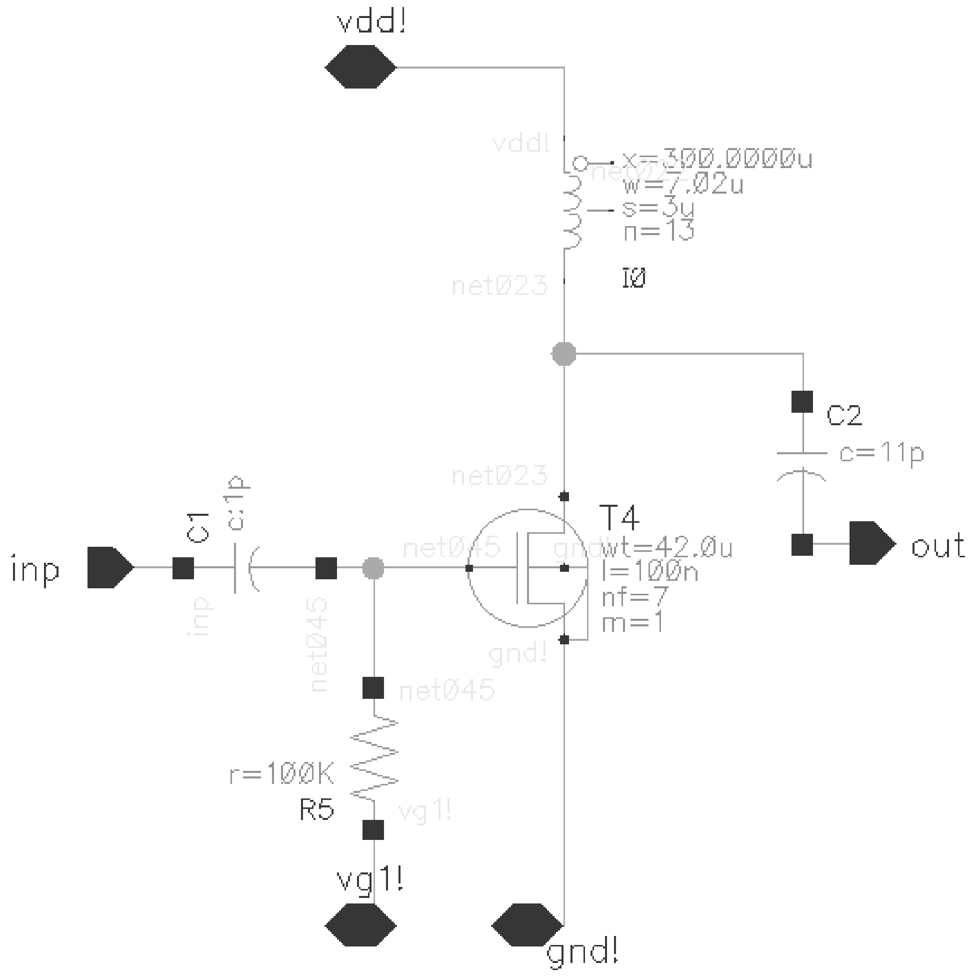


Figure 57: Class A Power Amplifier Cadence Schematic

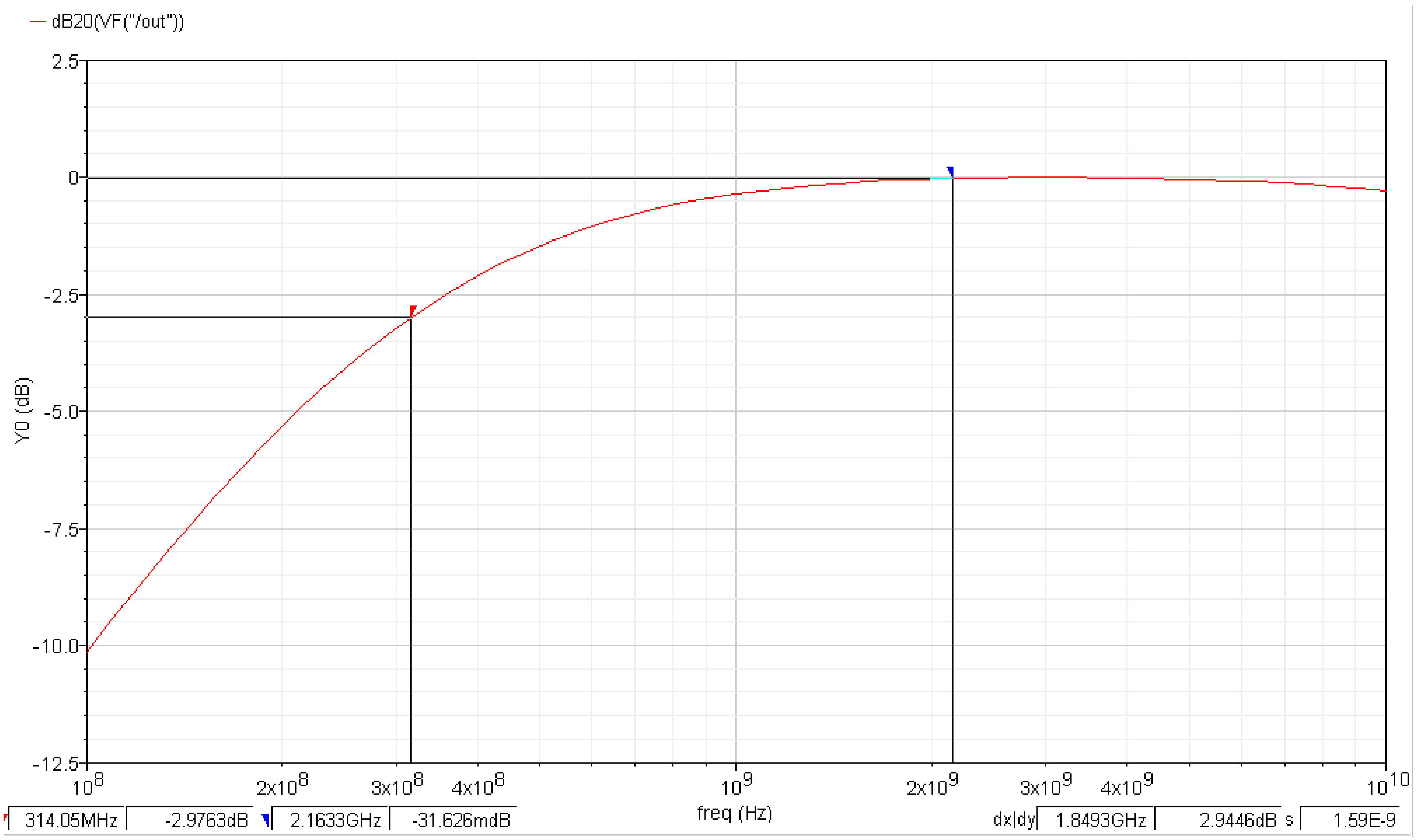


Figure 58: PA Frequency Response when driving a 50 Ω load

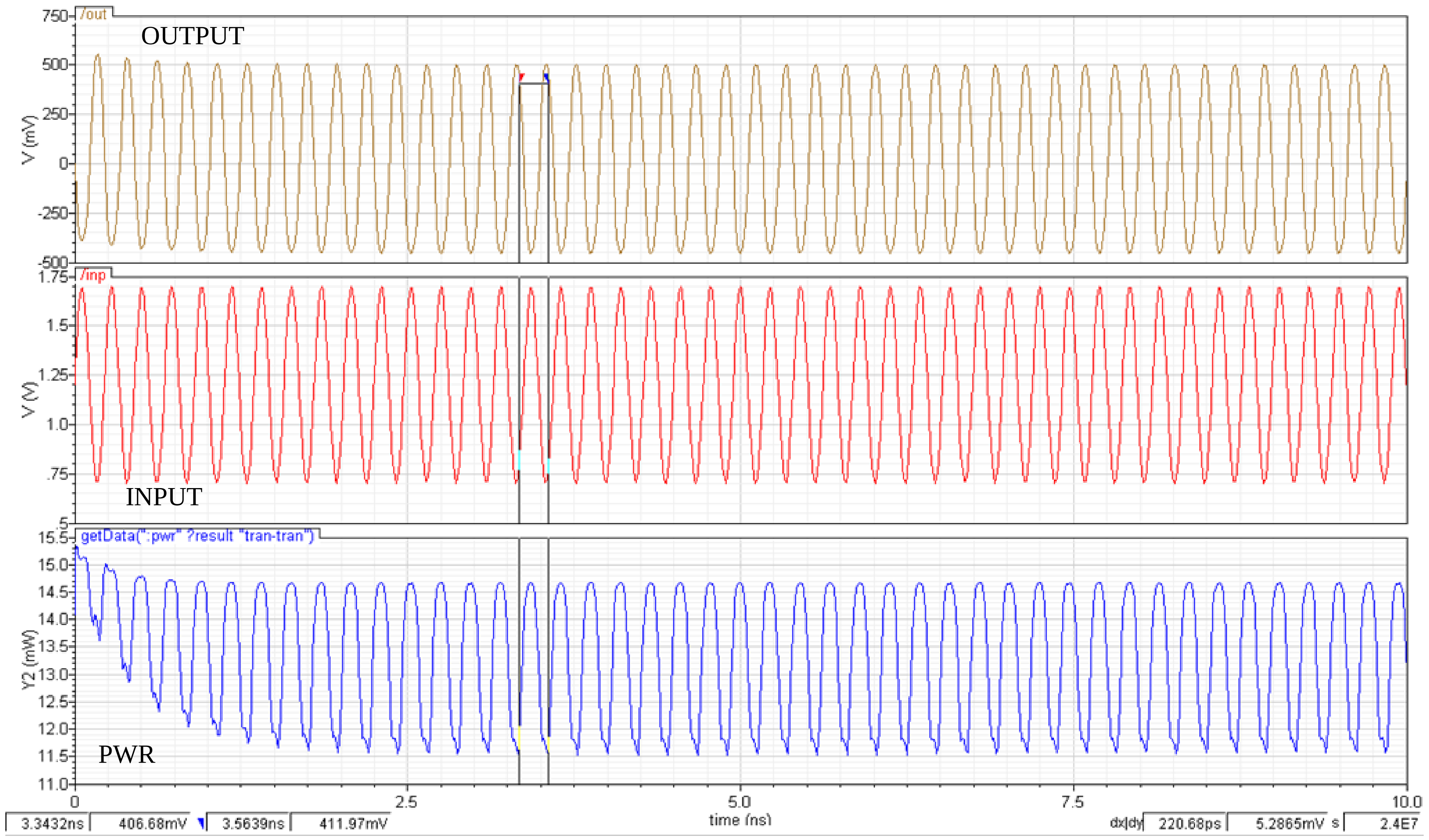


Figure 59: PA Sim. & PWR at 4.5 GHz, Avg PWR= 13 mW

III SIMULATION RESULTS

All the sub-components of the DDS discussed in the previous section were built in Cadence using 90nm CMOS technology. The schematic in Figure 60 shows the circuit of the Direct Digital Synthesizer which is used for the final simulation. The circuit contains a 16-bit phase accumulator, 16 registers, a digital buffer cell consisting of four stages inverters, a three-stage staggered-tuned cascode filter, an output driver and a 50 Ω load.

Missing from the complete DDS system is the multiplexer cell, de-multiplexer cell and the decoder cell that select the analog filter.

Only one analog filter from the filter bank, tuned for one operating frequency and bandwidth, was built and used in the simulation to show the conceptual operation. The analog filter bank can be built from copies of the presented filter tuned at different frequency bands. Initially designed to operate at 10 GHz, the clock frequency has to be reduced down to 9.09 GHz (Period: 0.11ns) for proper operation.

The input word used for the simulation is 8000h (32768 in decimal) to output a 4.54 GHz sine wave. Only the MSB output of the phase accumulator is needed for that frequency. The inputs of the phase accumulator are hard wired to VDD and GND to facilitate changing the value in Cadence. In a real circuit, the inputs of the phase accumulator are connected to other digital logic circuits.

The digital buffer is needed to boost the driving capability of the register cell because of the large input capacitance of the analog filter.

The analog filter and the output driver use biasing DC voltages, V1 and V2, set at 0.45, and 0.95 accordingly. It’s useful to note that to turn OFF the analog filter or the output driver; one could just set V1 and V2 to zero, whereas in a passive filter bank, the filter has to be disconnected from the circuit.

The simulation results are shown in Figure 61 and Figure 62. The following waveforms were captured:

1. DDS Output (1st from top)
2. Analog Filter output (2nd down)
3. Phase Accumulator output (3nd down)
4. Clock (bottom)

First, from the simulation shown in Figure 61, it can be seen that the output of the phase accumulator has a latency of 16 clock cycles.

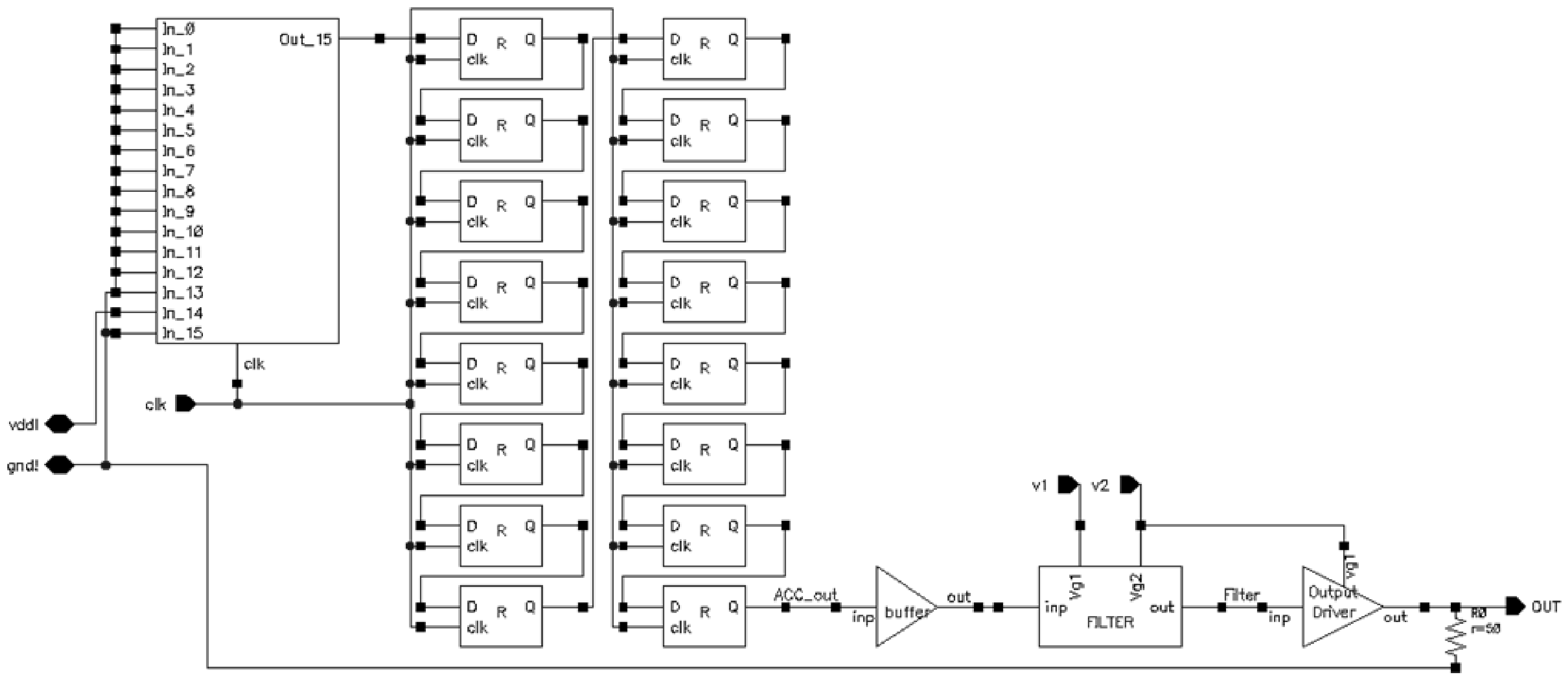


Figure 60: 16-bit DDS Schematic

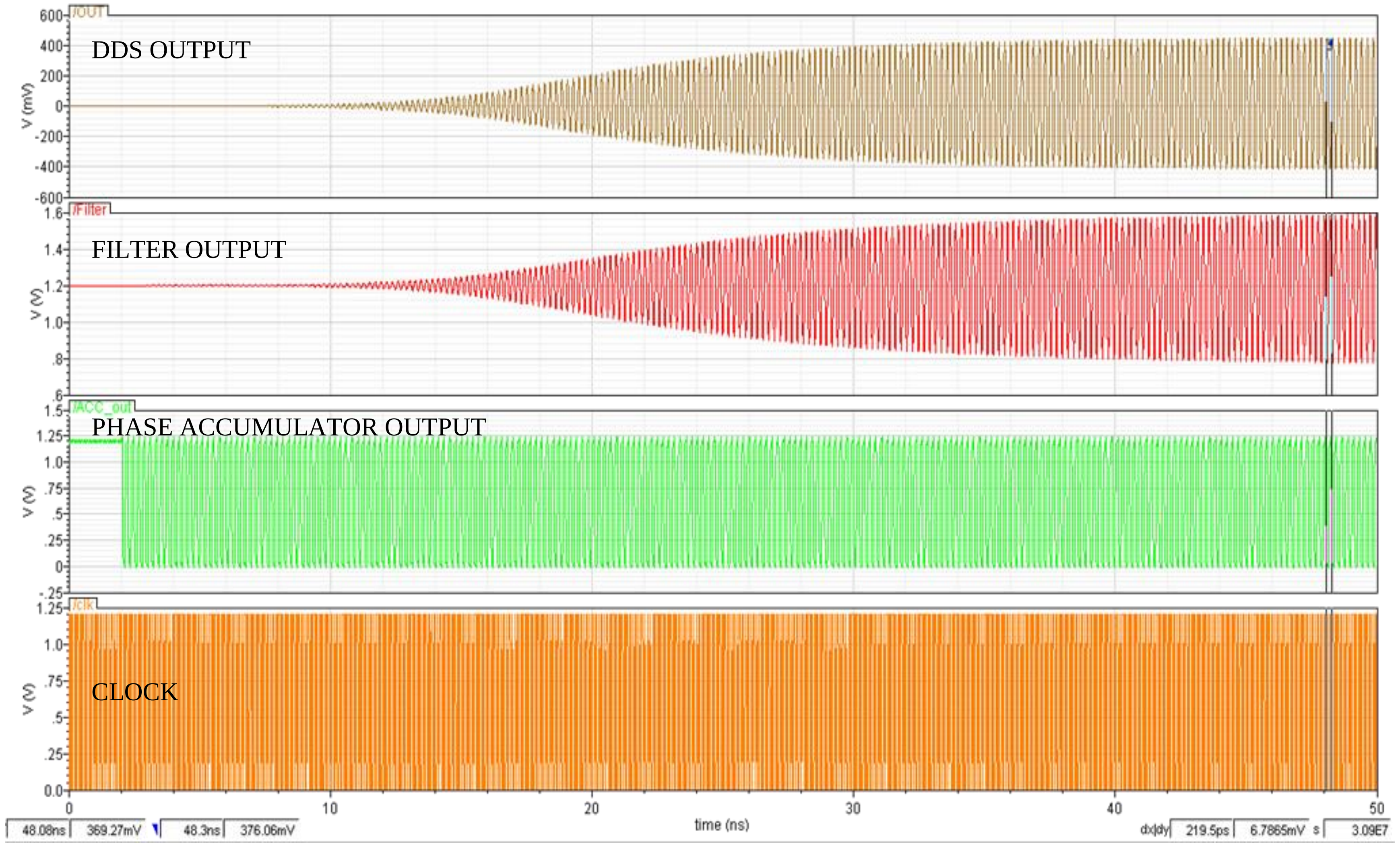


Figure 61: 16-bit DDS Simulation, M=8000h

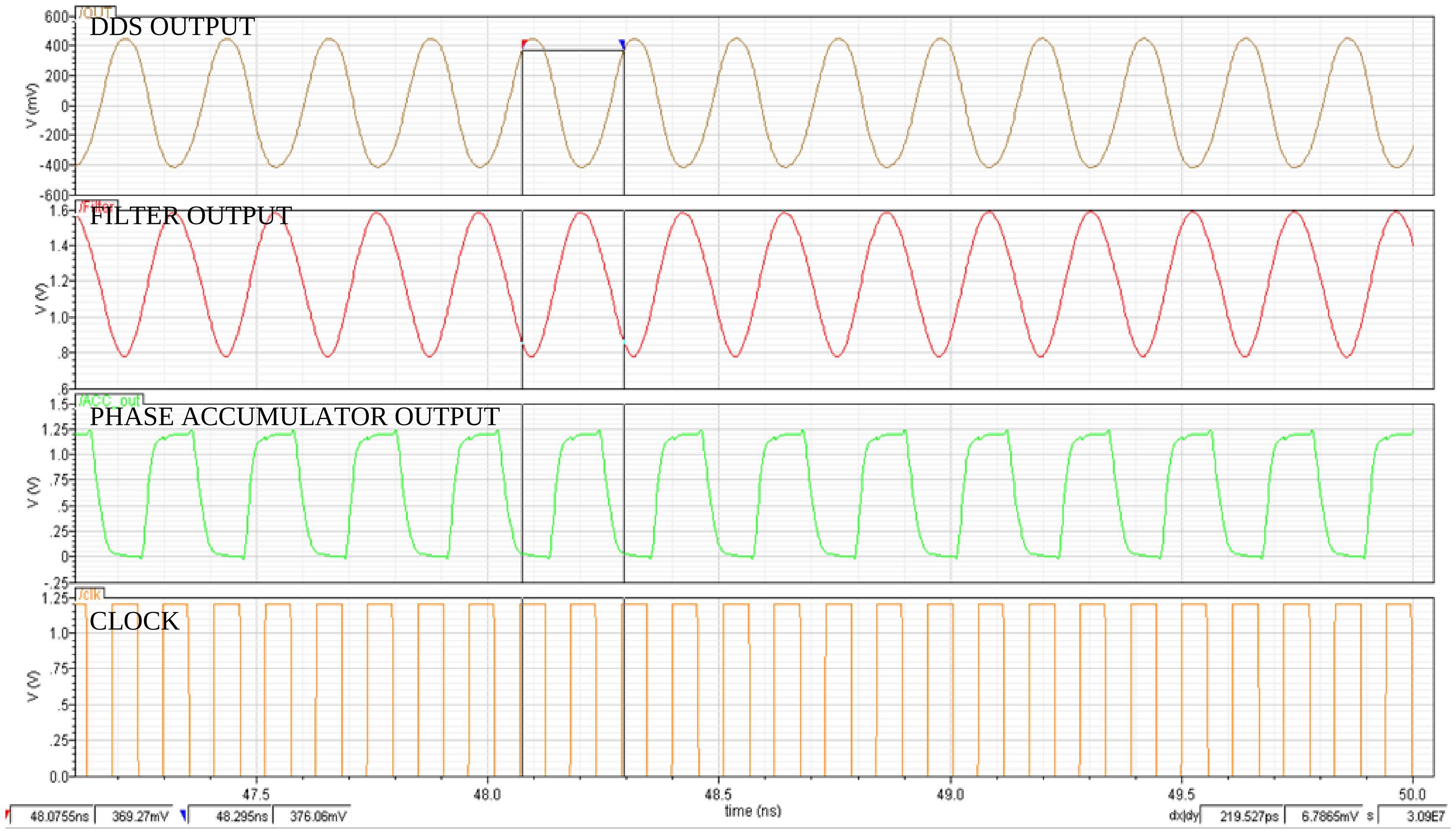


Figure 62: 16-bit DDS Simulation (Zoom In)

The output of analog filter stabilizes after approximately 45 ns. In Figure 62, the zoomed-in input waveform of the analog filter coming from the phase accumulator is a square wave with an amplitude of 1.2 V. The output of the filter is a sine wave with a peak-to-peak amplitude of 800mV and a DC voltage offset of 1.2 V. The output of the DDS is a sine wave, with 0 VDC, and with a voltage swing of $\pm$ 400 mV.

With a frequency tuning input word of 8000h (decimal=32768), the measured DDS output frequency is approximately 4.555 GHz.

The power consumption of the simulated DDS is captured in Figure 63. The entire DDS system consumes in average 27.72 mW with the selected input word of 8000h and clock frequency of 9.09 GHz. The peak power is approximately 125 mW.

The simulation is run again with different input words: 4000h (decimal=16384), and C000h (decimal=49152).

Figure 64 and Figure 65 shows the result with input words 4000h (2.5 GHz) and C000h (7.5 GHz). The outputs of the DDS for both inputs are reduced to 15 mVpp; a 36.5 dB attenuation.

Figure 66 and Figure 67 capture the 2nd and 3rd harmonics of the DDS system. The peak noise of the 2nd harmonics is approximately 27 dB down from the center frequency and the noise of the 3rd harmonics is 30 dB down from the center.

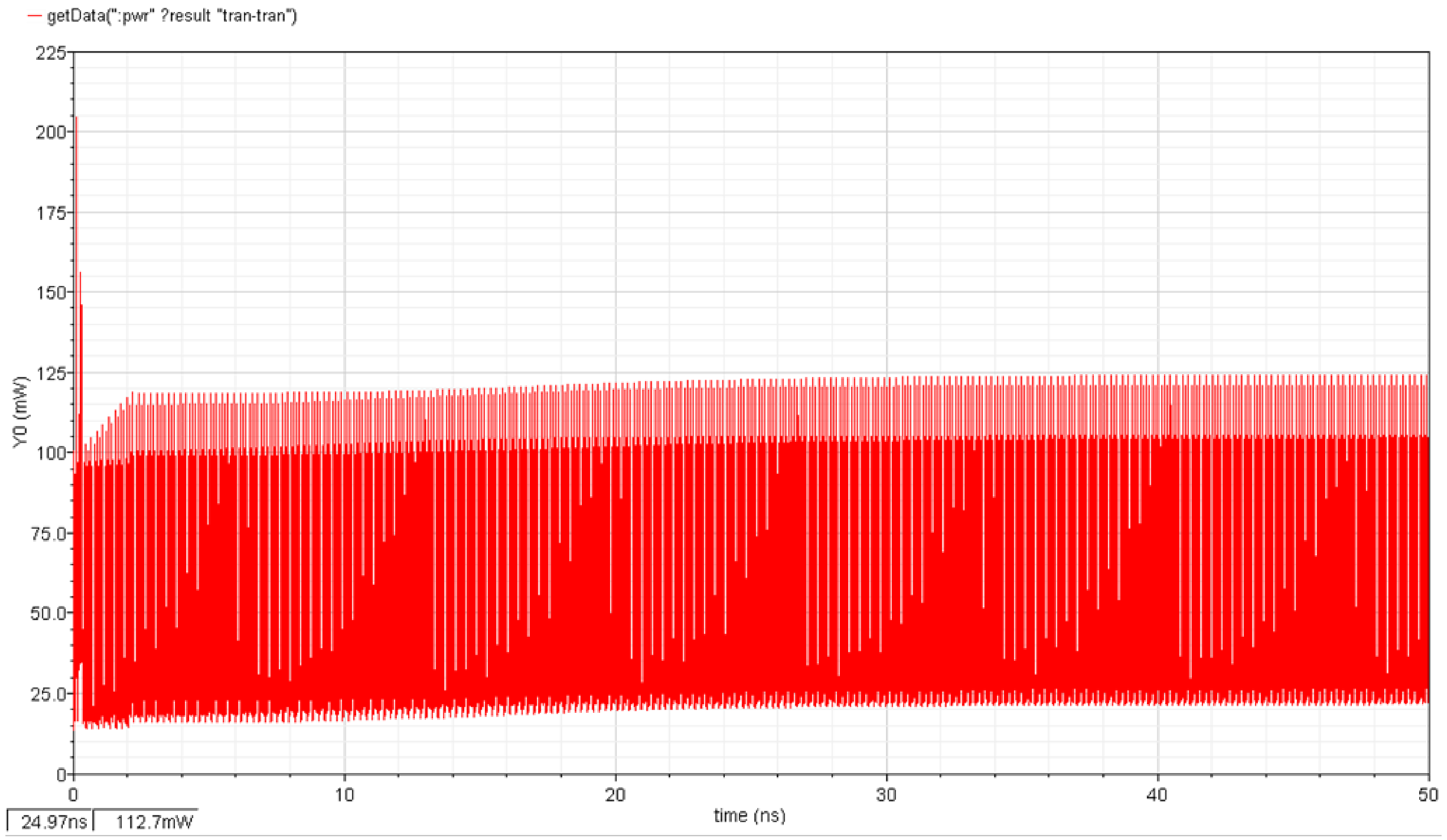


Figure 63: 16-bit DDS PWR consumpt., Avg PWR= 27.72mW

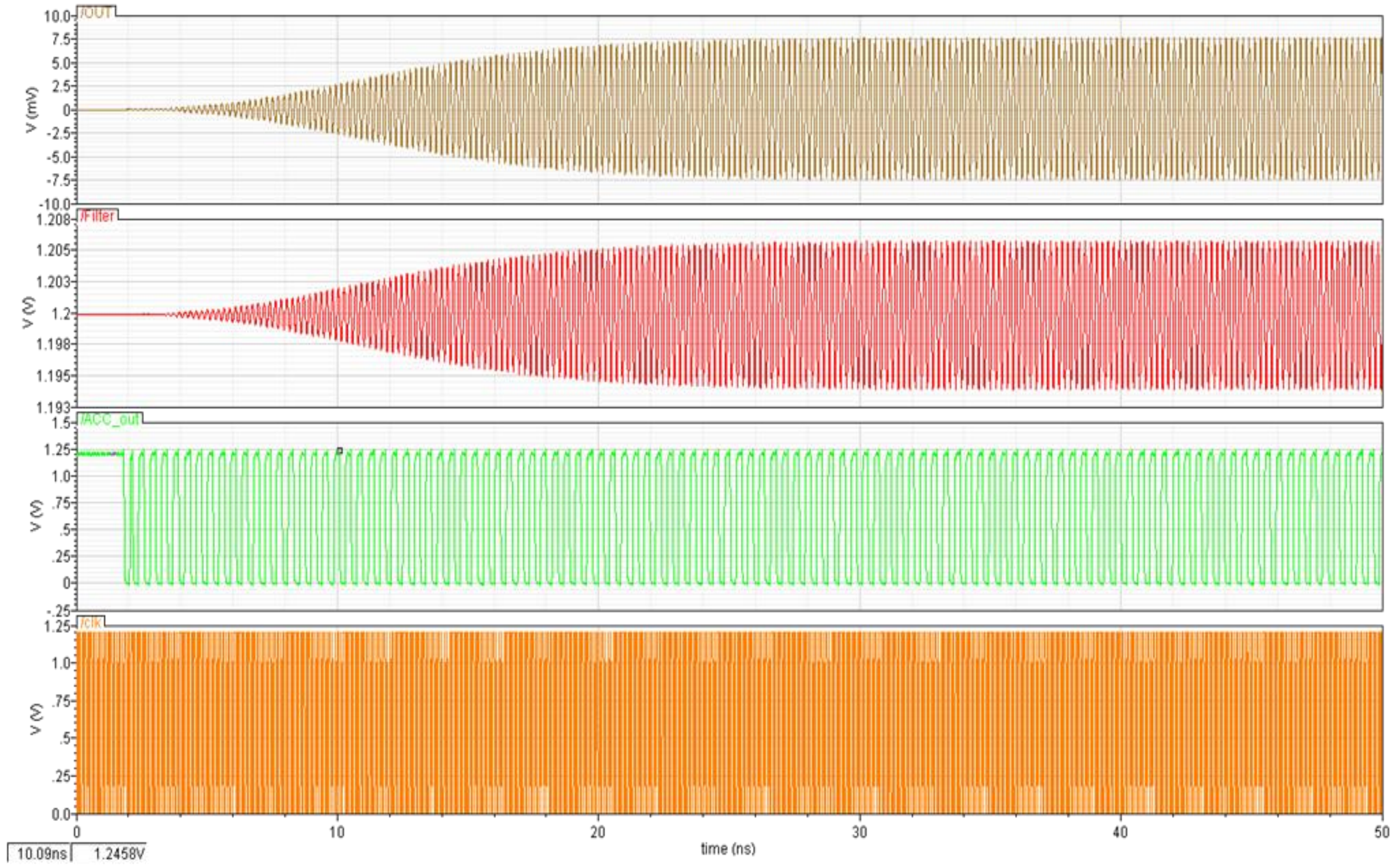


Figure 64: WORD=4000h, DDS Output 36.5 dB Attenuation

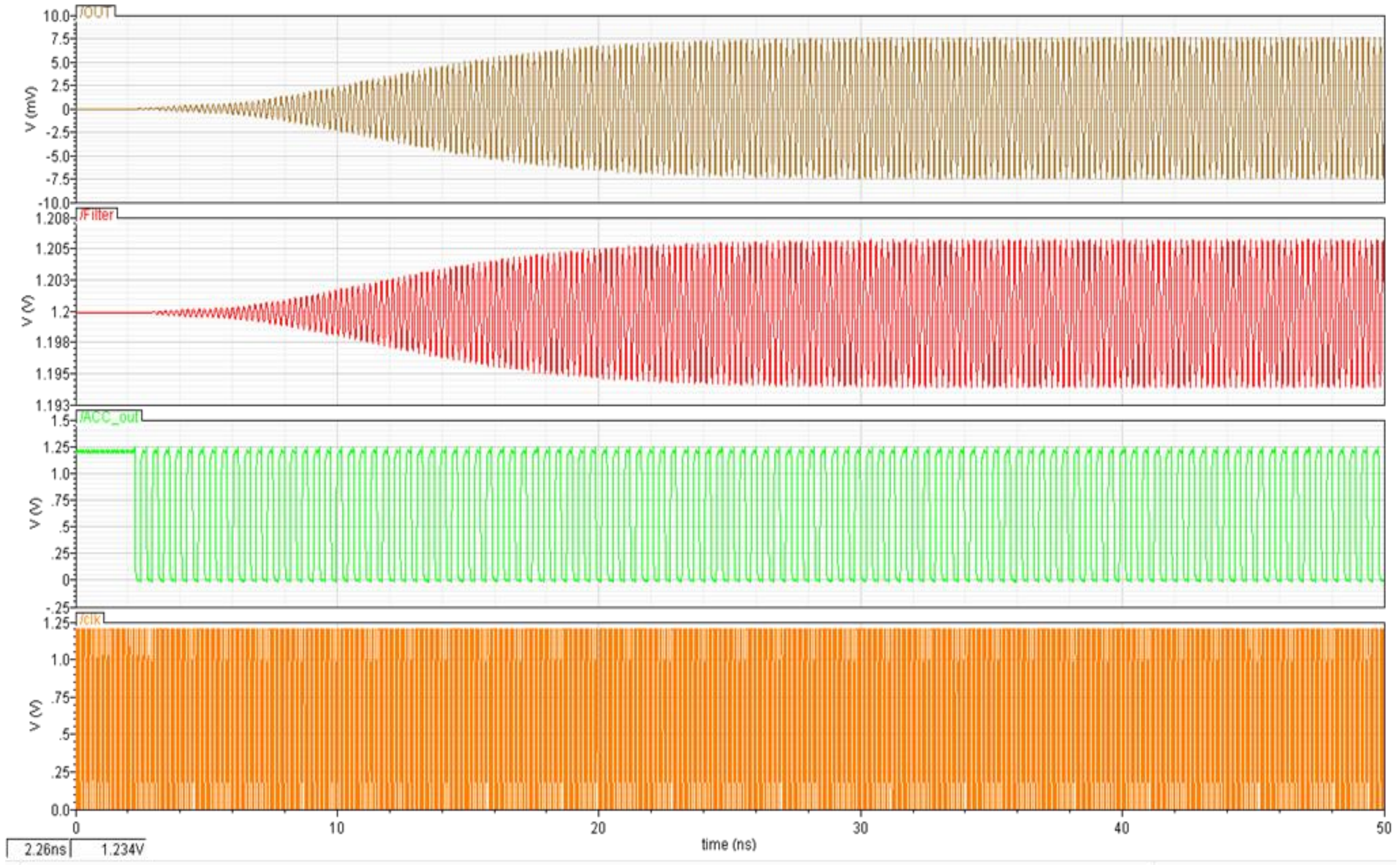


Figure 65: WORD=C000h, DDS Output 36.5 dB Attenuation

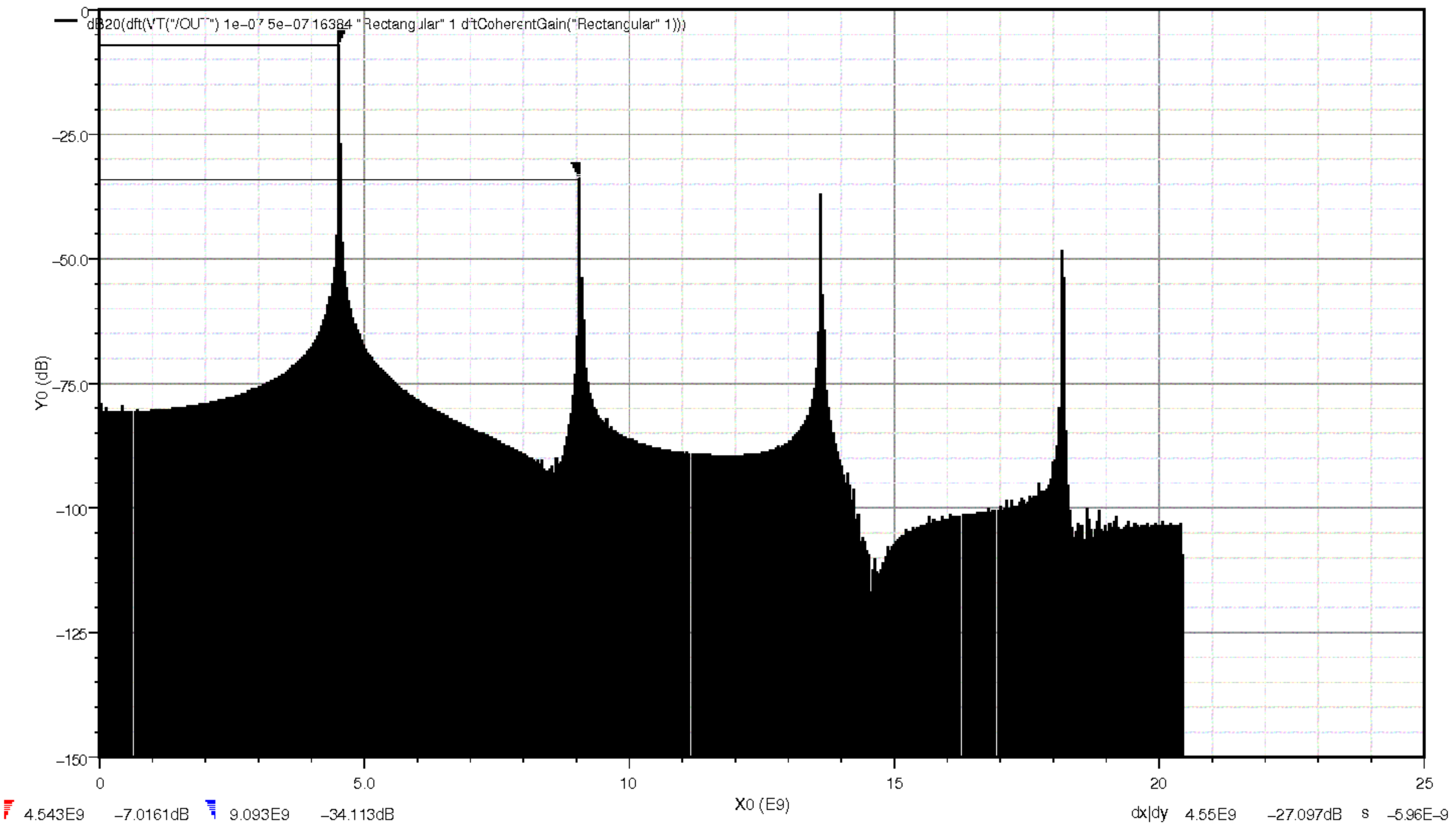


Figure 66: Spurious Free Dynamic Range, 2nd Harmonic

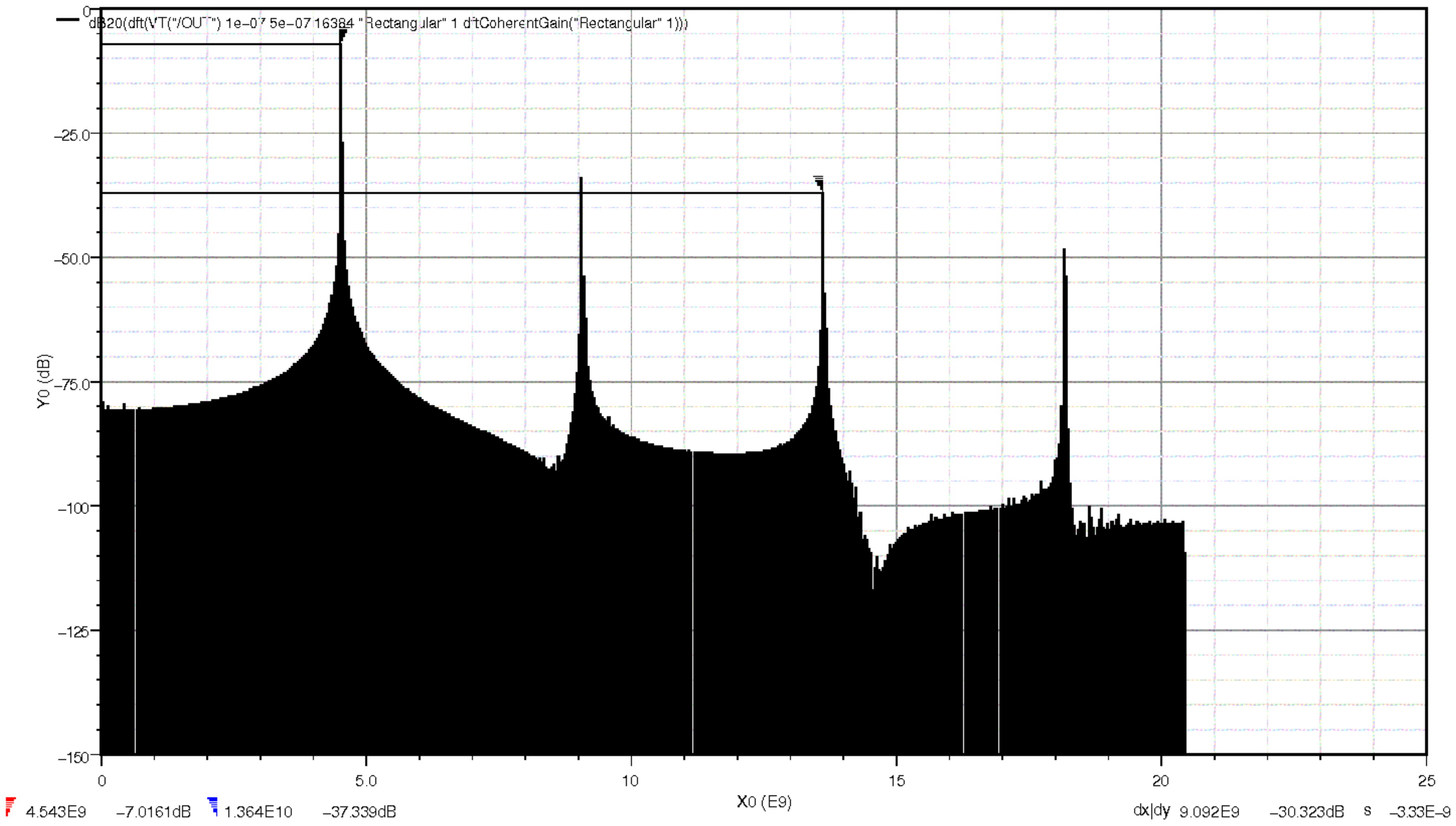


Figure 67: Spurious Free Dynamic Range, 3rd Harmonic

IV SUMMARY & CONCLUSION

In order to upgrade the DDS technology for use in wireless application, several changes were done to the DDS architecture. High speed optimizations techniques were applied to both digital and analog circuits to permit integration of all building blocks with 90 nm CMOS technology.

A new design of full-adder was proposed for the phase accumulator. The design concept of the full-adder consists of separating and paralleling the computation of the sum and the carry-out to reduce delay. Static logic mixed with Pass-Transistor logic was a way to reduce power consumption while operating at high speed. The operating speed of the adder is 10 GHz, with a propagation delay of approximately 22 ps.

Pipelining allows the phase accumulator to operate at high speed independently from the number of bit accuracy. RC delay optimization, using digital buffers, reduces the delay in the feedback line of the phase accumulator. The 16-bit phase accumulator operates at 9.09 GHz

High frequency analog circuits, re-using old RF technology, the tuned amplifiers, take advantage of the integration of the on-chip passive components to achieve high gain at high speed. The short bandwidth of the tuned amplifiers can be overcome by using multi-stage stagger-tuning amplifiers. The attenuation of the filter outside the operating bandwidth is approximately 36.5 dB. Higher attenuation can be achieved with a shorter operating bandwidth.

The output driver, a single ended class-A power amplifier, was used to drive the external load to obtain the best linearity. Alternatively, a differential power amplifier using the same family of transistor can be used to produce more output power.

The entire chain DDS with only one filter consumes approximately 30 mW. The measured settling time is about 45 ns. And the Free Spurious Dynamic Range is measured at 27 dB.

The proposed DDS design is feasible for use in wireless applications. All building blocks can be integrated with the low cost 90 nm CMOS technology. The design offers high operating frequency, low power consumption, and relatively low noise and spurs.

V FUTURE WORK

DDS technology is an interesting field that could offer many advantages to wireless communication. Although there are drawbacks in the architecture of the DDS, many published works have been focused on upgrading the capability of the DDS.

The replacement of the phase to amplitude converter by analog filters eliminates the need for a DAC and a ROM LUT. The outcome has the benefits of reducing power, increasing the operating frequency, and avoiding the spurs and errors associated with the DAC and ROM LUT.

Future works can include the design of the missing building blocks to the proposed architecture, which were not discussed in this thesis: the frequency control word decoder, the multiplexer and the de-multiplexer. These blocks are used to switch the analog filter bank.

Also not discussed in this thesis is the reference clock which can be generated from an on-chip phase lock-loop.

Other future work could consist of designing a differential power amplifier so that the DDS can deliver more output power to the external load with a supply rail voltage of 1.2 V.

VI REFERENCES

[1] Keliu Shu and Edgar Sanchez-Sinencio, CMOS PLL Synthesizers: Analysis and Design, Springer, 2005.

[2] J.M. Yang, S.Y. Kim, S. J. Kim, B. T. Jeon, "Fast Switching Frequency Synthesizer Using Direct Analog Techniques for Phase-Array Radar", Agency for Defencse Dev., Daejeo, Radar 97(Conf. Publ. No449), p386, 1997.

[3] David J. Foley, Michael P. Flynn, "CMOS DLL-Based 2-V 3.2-ps Jitter 1-GHz Clock Synthesizer and Temperature-Compensated Tunable Oscillator", IEEE Journal of Solid-State Circuits, vol. 36, NO. 3, p417, Mar. 2001.

[4] Ken Gentile, David Brandon, Ted Harris, "A Technical Tutorial on Digital Signal Synthesis", Analog Devices Inc, 1999.

[5] Sunita Khilar, Kiran Parmar, Saumi S, Dr. K. S. Dasgupta, "Design and Analysis of Direct Digital Frequency Synthesizer", First International Conference on Emerging Trends in Engineering and Technology, published in IEEE Computer Society,p1302-1306, 2008.

[6] Jouko Vankka, Direct Digital Synthesizers: Theory, Design and Applications, 2000.

[7] V. S. Reinhardt, “Spur Reduction Techniques in Direct Digital Synthesizers”, Proceedings of the 47th Frequency Control Symposium (IEEE/ERADCOM, Salt Lake City), 1993.

[8] Yingtao Jiang, Abdulkarim Al-Sheraidah, Yuke Wang, Edwin Sha, and Jin-Gyun Chung, “A Novel Multiplexer-based Low power Full Adder”, IEEE Transactions on Circuits and Systems—II: Express Briefs, VOL. 51, NO. 7, p345-348 July 2004.

[9] Amir Ali Khatibzadeh, Kaamran Raahemifar, “A Study and Comparison of Full-adder Cells Based on the Standard Static CMOS Logic”, Canadian Conference on Electrical and Computer Engineering, p2139-2142, May 2004.

[10] N. Weste, D. Harris, A Banerjee, CMOS VLSI Design: A Circuits and Systems Perspective, Pearson Education, 3rd Edition, p304-309, p47, 2005.

[11] David J. Betowski and Valeriu Beiu, “Considerations for Phase Accumulator Design for Direct Digital Frequency Synthesizers”, IEEE Int. Conf. Neural Networks & Signal Processing Nanjing China, p176-179, December 2003.

[12] Rene J. Romero-Troncoso, Guillermo Espinosa-Flores-Verdad, “Algorithm for Phase Accumulator Synthesis For Applications in DDS”, IEEE , p210-213, 1999.

[13] Sodagar A. M., Lahiji G.R., A pipelined ROM-Less Architecture for Sine-Output Direct Digital Frequency

Synthesizers Using the Second Order Parabolic Approximation, IEEE Trans., Circuit and Systems TI. Vol.48, p850-857, Sept. 2001.

[14] Sedra and Smith, Microelectronic Circuits, Oxford University Press, 1998.

[15] Duran Leblebici, Yusuf Leblebici, Fundamentals of High-Frequency of CMOS Analog Integrated Circuits, 2009.

[16] Christopher Bowick, RF Circuit Design, Newnes, 1982.

[17] Bram Nauta, Analog CMOS Filter for Very High Frequencies, Kluwer Academics Publisher, 1993.

[18] Yichuang Sun, "Wireless Communication Circuits and Systems", Institution of Electrical Engineers, 2004.

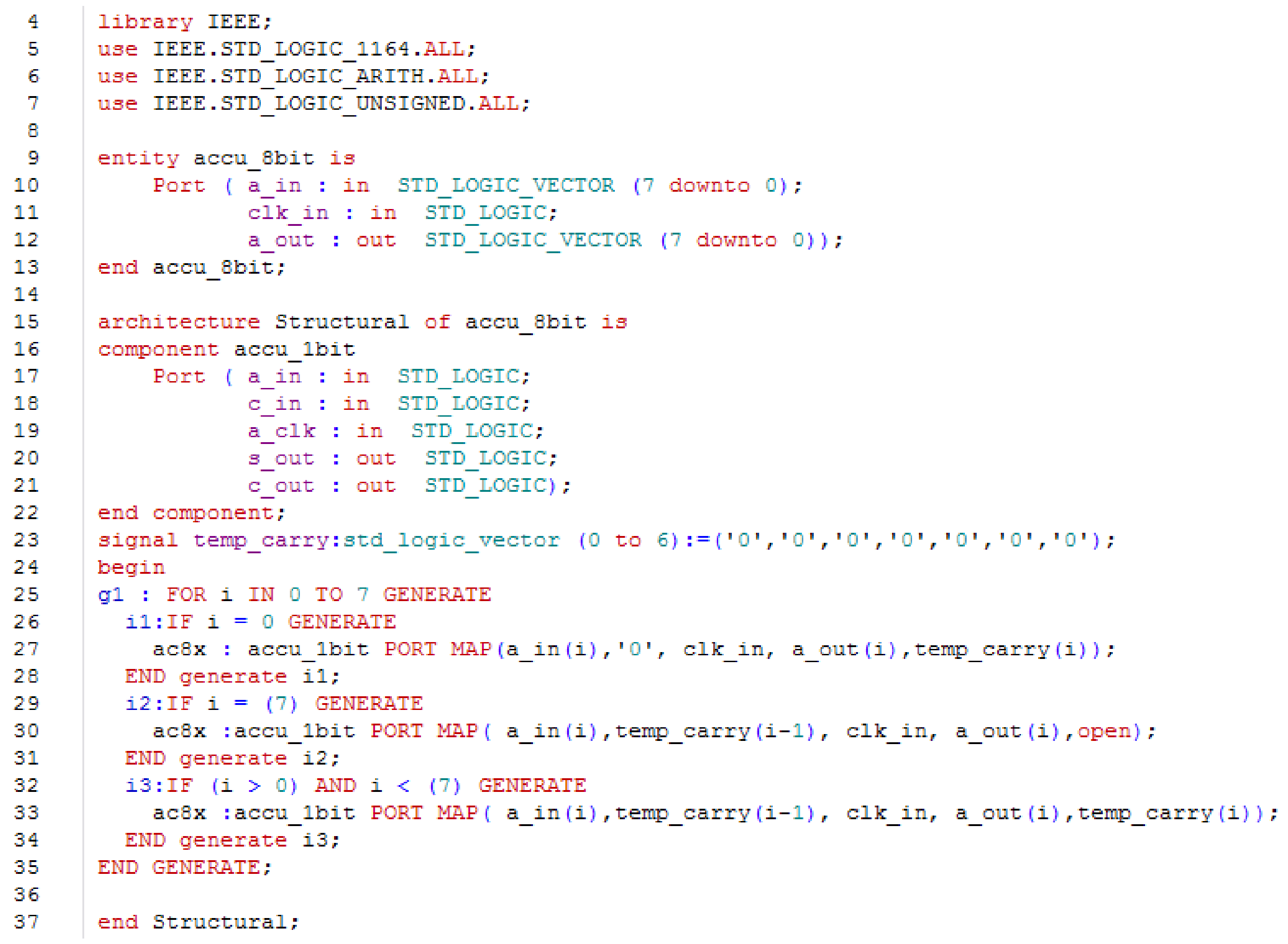
[19] Thomas H Lee, Design of CMOS Radio-Frequency, Cambridge University Press, 1998.

[20] Behzad Razavi, Design of Analog Integrated Circuit, McGraw Hill, 2001.

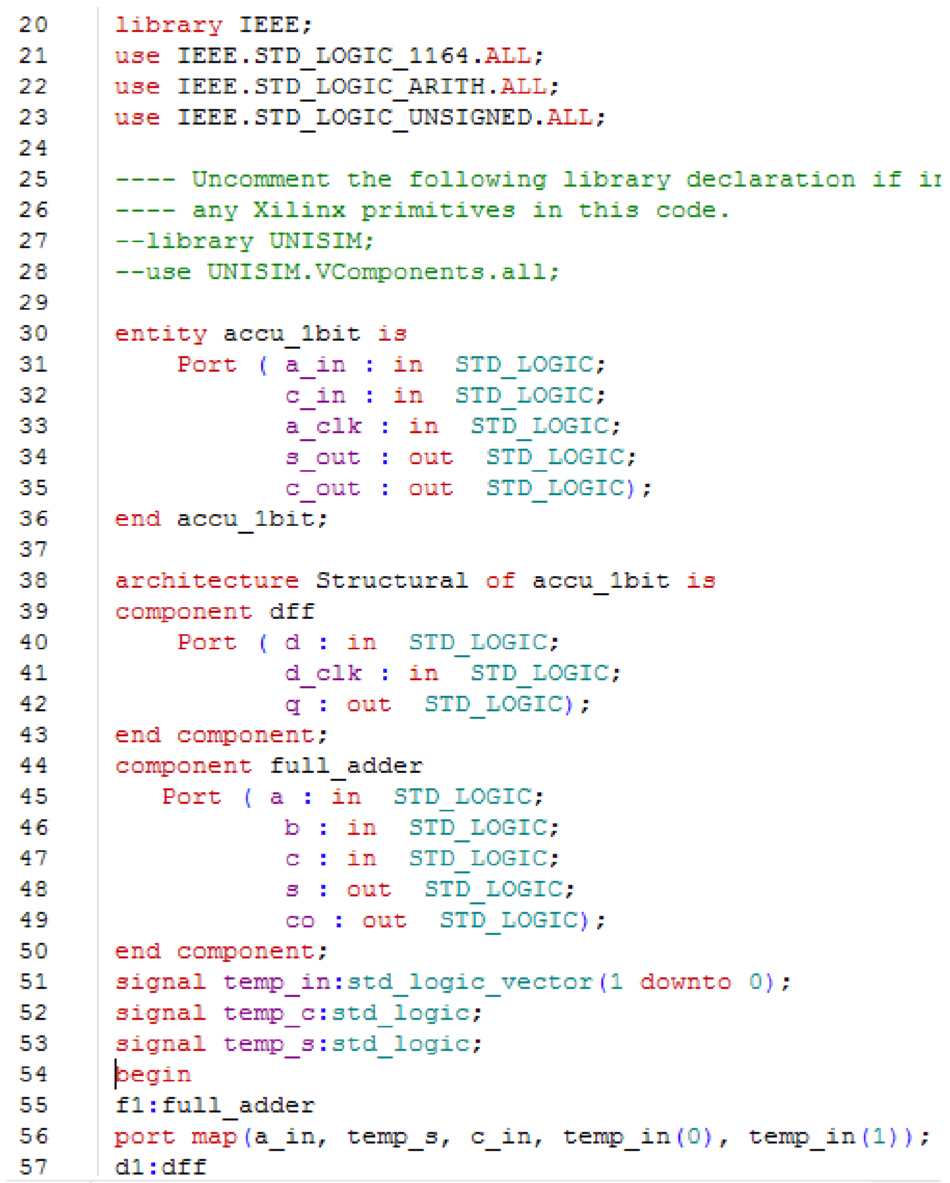
VII ATTACHMENT: 8-BIT DDS VHDL FILES

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instant
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity fre_sync is
    Port ( inputs : in  STD_LOGIC_VECTOR (7 downto 0);
           f_clk : in  STD_LOGIC;
           outputs : out  STD_LOGIC_VECTOR (7 downto 0));
end fre_sync;

architecture Structural of fre_sync is
component accu_8bit
    Port ( a_in : in  STD_LOGIC_VECTOR (7 downto 0);
           clk_in : in  STD_LOGIC;
           a_out : out  STD_LOGIC_VECTOR (7 downto 0));
end component;
component register_8bit
Port ( input : in  STD_LOGIC_VECTOR (7 downto 0);
           r_clk : in  STD_LOGIC;
           output : out  STD_LOGIC_VECTOR (7 downto 0));
end component;
component n_register
  Generic(n:natural);
    Port ( reg_in : in  STD_LOGIC;
           reg_clk : in  STD_LOGIC;
           reg_out : out  STD_LOGIC);
end component;
component dff
    Port ( d : in  STD_LOGIC;
           d_clk : in  STD_LOGIC;
           q : out  STD_LOGIC);
end component;
signal temp1: std_logic_vector(0 to 7);
signal temp2: std_logic_vector(0 to 7);
--signal carry: std_logic_vector(0 to 7);
signal sum: std_logic_vector(0 to 7);
--signal temp3: std_logic_vector(0 to 7);
begin
r1:register_8bit
```

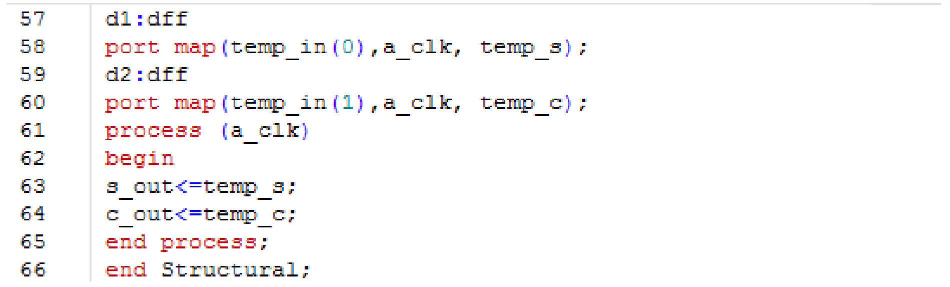
```
  port map(inputs,f_clk, temp1);
    --Pipeline in
h1 : FOR i IN 0 TO 7 GENERATE
  j1:IF i > (0) GENERATE
    shix :n_register
    generic map(i+1)
    PORT MAP(temp1(i),f_clk, temp2(i));
  END generate j1;
  j2:IF i = (0) GENERATE
    shix :dff
    PORT MAP( temp1(i),f_clk, temp2(i));
  END generate j2;
END GENERATE;
  --accumulator
a1:accu_8bit
  port map(temp2,f_clk,sum);
  --Pipeline out
g1 : FOR i IN 0 TO 7 GENERATE
  i1:IF i < (7) GENERATE
    shox :n_register
    generic map(8-i)
    PORT MAP( sum(i),f_clk, outputs(7-i));
  END generate i1;
  i2:IF i = (7) GENERATE
    shox :dff
    PORT MAP( sum(i),f_clk, outputs(i-7));
  END generate i2;
END GENERATE;
end Structural;
```



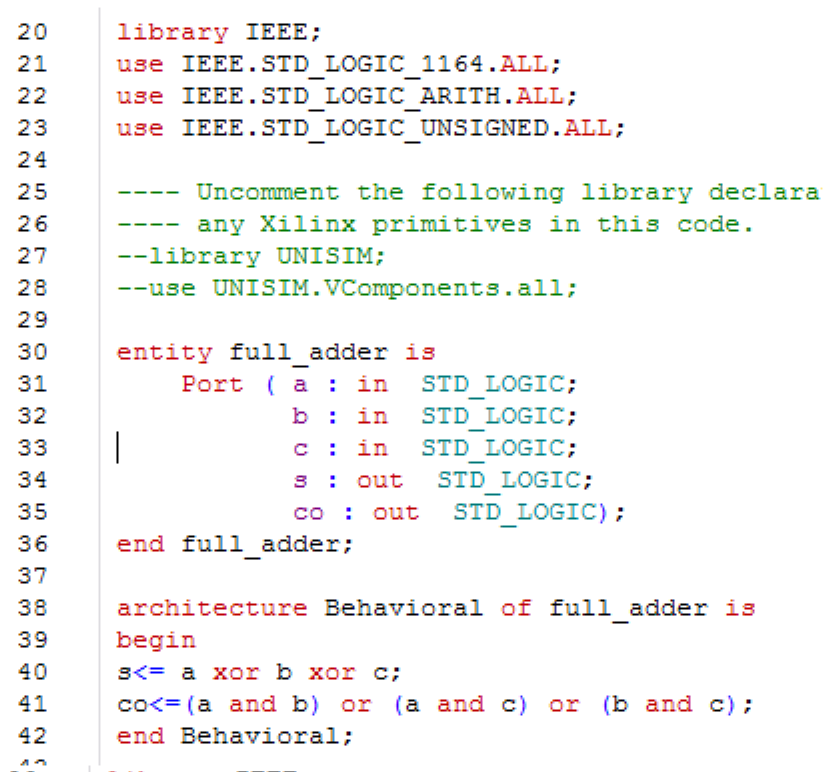
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity accu_8bit is
    Port ( a_in : in  STD_LOGIC_VECTOR (7 downto 0);
           clk_in : in  STD_LOGIC;
           a_out : out  STD_LOGIC_VECTOR (7 downto 0));
end accu_8bit;

architecture Structural of accu_8bit is
component accu_1bit
    Port ( a_in : in  STD_LOGIC;
           c_in : in  STD_LOGIC;
           a_clk : in  STD_LOGIC;
           s_out : out  STD_LOGIC;
           c_out : out  STD_LOGIC);
end component;
signal temp_carry:std_logic_vector (0 to 6):=('0','0','0','0','0','0','0');
begin
g1 : FOR i IN 0 TO 7 GENERATE
  i1:IF i = 0 GENERATE
    ac8x : accu_1bit PORT MAP(a_in(i),'0', clk_in, a_out(i),temp_carry(i));
  END generate i1;
  i2:IF i = (7) GENERATE
    ac8x :accu_1bit PORT MAP( a_in(i),temp_carry(i-1), clk_in, a_out(i),open);
  END generate i2;
  i3:IF (i > 0) AND i < (7) GENERATE
    ac8x :accu_1bit PORT MAP( a_in(i),temp_carry(i-1), clk_in, a_out(i),temp_carry(i));
  END generate i3;
END GENERATE;

end Structural;
```



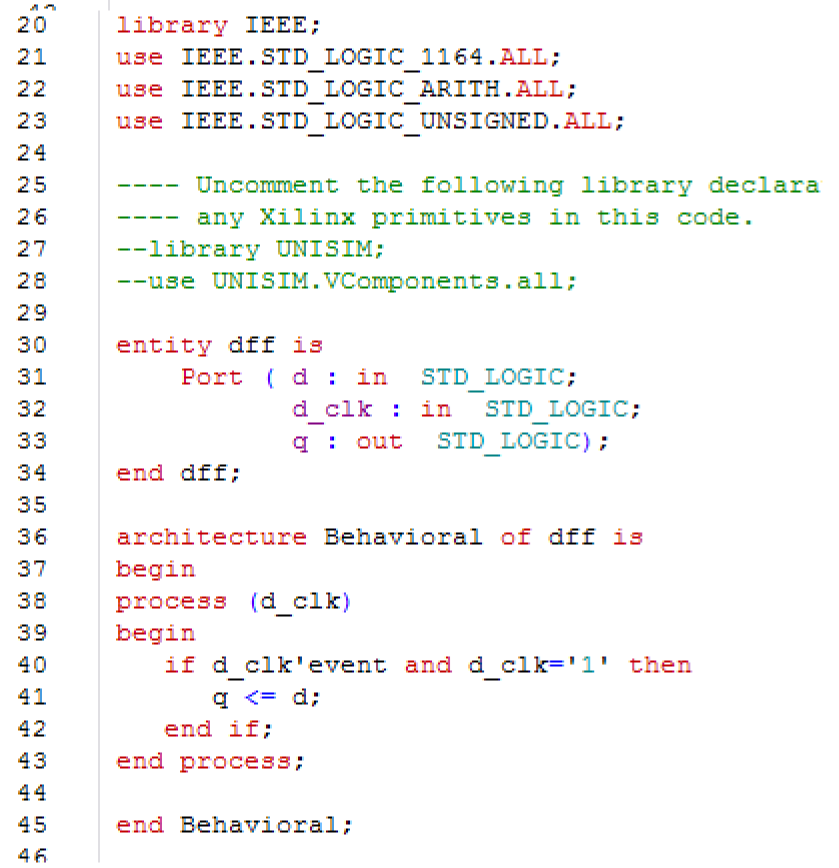
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if i
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity accu_1bit is
    Port ( a_in : in  STD_LOGIC;
           c_in : in  STD_LOGIC;
           a_clk : in  STD_LOGIC;
           s_out : out  STD_LOGIC;
           c_out : out  STD_LOGIC);
end accu_1bit;

architecture Structural of accu_1bit is
component dff
    Port ( d : in  STD_LOGIC;
           d_clk : in  STD_LOGIC;
           q : out  STD_LOGIC);
end component;
component full_adder
   Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           c : in  STD_LOGIC;
           s : out  STD_LOGIC;
           co : out  STD_LOGIC);
end component;
signal temp_in:std_logic_vector(1 downto 0);
signal temp_c:std_logic;
signal temp_s:std_logic;
begin
f1:full_adder
port map(a_in, temp_s, c_in, temp_in(0), temp_in(1));
d1:dff
```



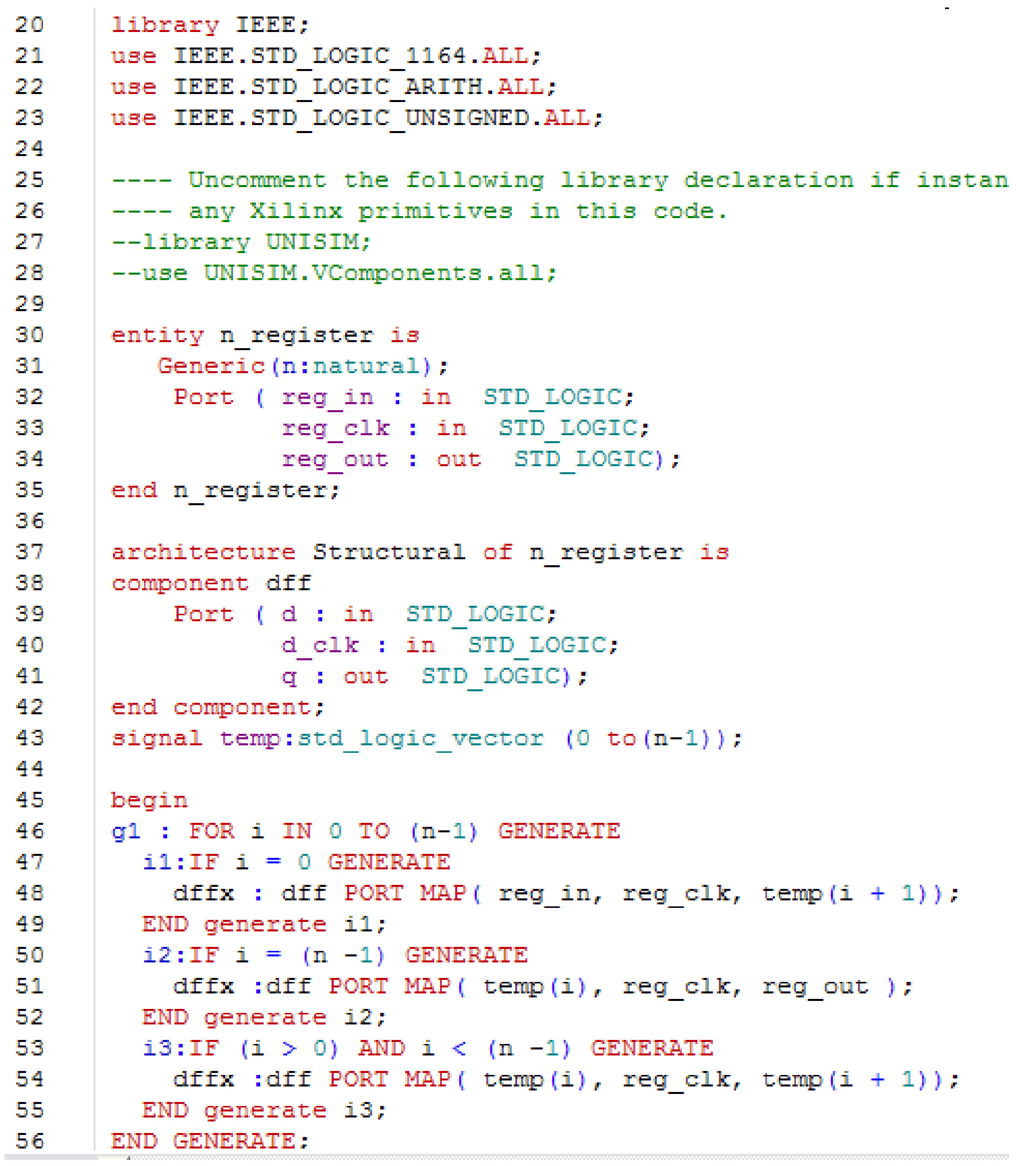
```vhdl
d1:dff
port map(temp_in(0),a_clk, temp_s);
d2:dff
port map(temp_in(1),a_clk, temp_c);
process (a_clk)
begin
s_out<=temp_s;
c_out<=temp_c;
end process;
end Structural;
```



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declara
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity full_adder is
    Port ( a : in  STD_LOGIC;
           b : in  STD_LOGIC;
           c : in  STD_LOGIC;
           s : out  STD_LOGIC;
           co : out  STD_LOGIC);
end full_adder;

architecture Behavioral of full_adder is
begin
s<= a xor b xor c;
co<=(a and b) or (a and c) or (b and c);
end Behavioral;
```



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declara
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity dff is
    Port ( d : in  STD_LOGIC;
           d_clk : in  STD_LOGIC;
           q : out  STD_LOGIC);
end dff;

architecture Behavioral of dff is
begin
process (d_clk)
begin
   if d_clk'event and d_clk='1' then
      q <= d;
   end if;
end process;

end Behavioral;
```



```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---- Uncomment the following library declaration if instan
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity n_register is
   Generic(n:natural);
    Port ( reg_in : in  STD_LOGIC;
           reg_clk : in  STD_LOGIC;
           reg_out : out  STD_LOGIC);
end n_register;

architecture Structural of n_register is
component dff
    Port ( d : in  STD_LOGIC;
           d_clk : in  STD_LOGIC;
           q : out  STD_LOGIC);
end component;
signal temp:std_logic_vector (0 to(n-1));

begin
g1 : FOR i IN 0 TO (n-1) GENERATE
  i1:IF i = 0 GENERATE
    dffx : dff PORT MAP( reg_in, reg_clk, temp(i + 1));
  END generate i1;
  i2:IF i = (n -1) GENERATE
    dffx :dff PORT MAP( temp(i), reg_clk, reg_out );
  END generate i2;
  i3:IF (i > 0) AND i < (n -1) GENERATE
    dffx :dff PORT MAP( temp(i), reg_clk, temp(i + 1));
  END generate i3;
END GENERATE;
```