A 40 – 100MHz PHASE-LOCKED LOOP FREQUENCY SYNTHESIZER WITH BUILT-IN SELF-TEST

A Thesis

by

Yankey Jephthah

Submitted to the Department of Computer Engineering

Kwame Nkrumah University of Science and Technology in partial fulfillment

of the requirements for the degree of

Master of Philosophy in Computer Engineering

November 2012

Declaration

I hereby declare that this submission, to the best of my knowledge, contains no material previously published nor material which has been accepted for the award of a degree by this or any other University, except where due acknowledgment has been given in the text.

YANKEY JEPHTHAH — DATE

(CANDIDATE)

DR. KWAME OSEI BOATENG — DATE

(SUPERVISOR)

DR. KWAME OSEI BOATENG — DATE

(HEAD OF DEPARTMENT)

Abstract

The Phase locked loop (PLL) is one of the most important devices in modern electronic systems. PLLs are widely used for clock generation or frequency synthesis in communication systems, computers, radio and other electronic applications. However, due to the use of expensive external equipment and amount of time involved, traditional VLSI testing methods are inefficient for testing of PLLs.

In this thesis, a fully functional PLL frequency synthesizer which operates from 40MHz to 100MHz is designed. The designed PLL exhibits phase noise of -71dBc/Hz at 1kHz, which is low enough for a wide array of applications.

To solve the testing problem, Built-In Self-Test (BIST) is employed. A BIST scheme based on a defect-oriented method of testing is proposed. A prototype adds BIST circuitry, a good part of which is derived from existing components of the original design. The PLL BIST scheme is generic and hence portable to similar PLL designs. One significant addition unit is a simple response collector that combines shifting and counting functionalities.

The entire system is designed in a typical CMOS process using a 3V power supply which is commonly found in today's portable products.

Spectre® simulations of the PLL show that it is capable of synthesizing any frequency between 40 and 100MHz within a reasonably short acquisition time. The output waveform of the generated signal is clean and shows no spikes whatsoever. Experimental simulations also reveal that the BIST circuitry is capable of generating the exact test pattern needed. It also performs efficiently all the unique checks which make up the PLL BIST. The final test output is very consistent and produces the same results for a number of different runs of the simulation.

Dedication

Dedicated to Dr. Kwame Osei Boateng. Thank you for being far more than just a supervisor.

Acknowledgements

I will most importantly like to thank my parents Mr. and Mrs. Yankey for their emotional and financial support all these years. I realize it couldn't always have been easy, and I'll be eternally grateful.

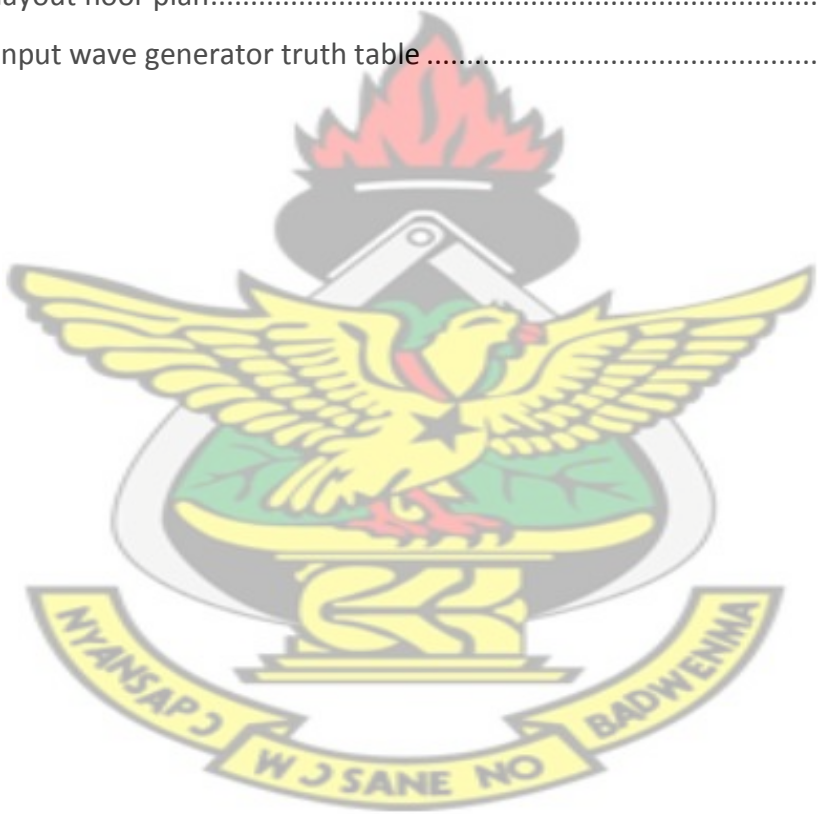
I will also like to thank Dr. Kwasi Diawuo for his assistance and guidance especially during the first year of my work. One person who deserves special mention is Roland Ribeiro. He took time to teach me how to use CADENCE efficiently. A lot of times, he would stop his own work to assist me with a problem I was having, and I feel truly indebted to him. I will also like to express my gratitude towards Eugene Foli and Richard Turkson for helping me with my design.

Finally, I will like to thank my siblings Huldah, Frances, and Paa Kwesi for the care and deep concern they demonstrated during this period.

Table of Contents

List of Figures

List of Tables



Chapter 1

Introduction

1.0 Background of Study

The Phase-locked loop, commonly known as the PLL is an essential component of modern electronic systems [1]. Having a wide range of applications over a broad frequency spectrum, the PLL has become one of the most essential elements in microprocessor boards of complex systems, wired and wireless communication systems and many other systems [2]. The fundamental PLL circuit has been around since the early 1920s as the earliest descriptions of the phase-locked loop appeared in papers by Appleton in 1923 and de Belliscize in 1932 [3].

While the first PLLs were realized with discrete components, they became available as ICs in about 1965 [4]. The first of these were analog devices which were modeled as linear circuits (LPLLs). A few years later, the first "digital" PLLs (DPLLs) became available, but these were actually analog systems with the phase detector built from logic circuits. Both LPLLs and DPLLs may be grouped together and referred to as "mixed signal PLLs". This simplifies analysis as both classes may now be analyzed with one theory.

The basic PLL circuit which has changed very little since its creation synchronizes an output signal with an input reference signal by comparing the phase difference between the two. The PLL works to produce an output signal that has the same frequency as the input reference signal and also a constant phase difference. A block diagram of this PLL is shown in Figure 1.1.

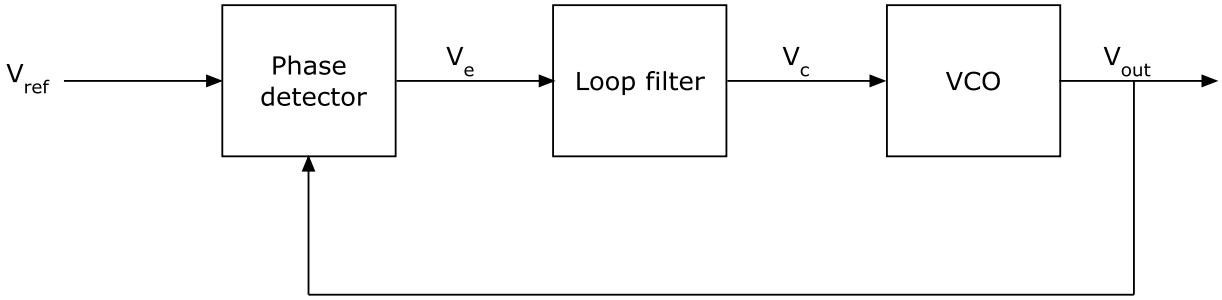


Figure 1.1 - Basic PLL Block Diagram

Advances in integrated circuit technology allowed phase-locked loop circuits to be used commonly in many areas such as communications, wireless systems, consumer electronics, and motor control [5]. The phase-locked loop is used in motor control to synchronize the motor speed to a reference frequency with extreme accuracy [3]. The phase-locked loop is used in the area of consumer electronics for applications ranging from television sets to microprocessors. The phase-locked loop performs the horizontal and vertical synchronization and color subcarrier reconstruction in television sets [4]. The phase-locked loop is used in microprocessors and other digital circuits to generate a low jitter clock signal [6]. Typical communications applications of PLLs include clock and data recovery, coherent demodulation of amplitude, frequency, and phase-modulated signals, phase-locked loop receivers, and frequency synthesis [5].

Such a vast array of applications has made the PLL quite a popular circuit among today's engineers. Like almost every physical system though, PLLs have to be constantly tested for correct operation and reliability. Classic VLSI testing methods have in the past been used in the testing of PLLs. However, these testing methods have proved expensive and time consuming. To solve this problem, a lot of research has been done in the use of built-in self-test (BIST) in VLSI circuits, including PLLs [7], [8], [9] [10], [11] .

The concept of built-in self-test, commonly abbreviated BIST, in its simplest form, involves the design of a circuit with the ability to test itself to determine whether it is "good" or "bad" (fault-

free or faulty, respectively). This typically requires that additional circuitry and functionality be incorporated into the original design of the circuit to facilitate the self-testing feature.

The additional circuitry must be capable of generating test patterns as well as provide a mechanism to determine if the output responses of the circuit under test (CUT) to the test patterns correspond to that of a fault-free circuit [12].

Figure 1.1 shows a typical implementation of BIST circuitry. In this BIST architecture, the test pattern generator (TPG) and output response analyzer (ORA) may be considered as the essential blocks. The response analyzer is also referred to as the response collector. While the TPG produces a sequence of patterns for testing the CUT, the ORA compacts the output responses of the CUT into a format that can be used to make a 'faulty/faultless' decision.

Aside from the normal system I/O pins, the incorporation of BIST may also require additional I/O pins for initializing the BIST sequence (the BIST Start control signal), and indicating BIST results. Sometimes an optional pin to indicate that the BIST sequence is complete may be included.

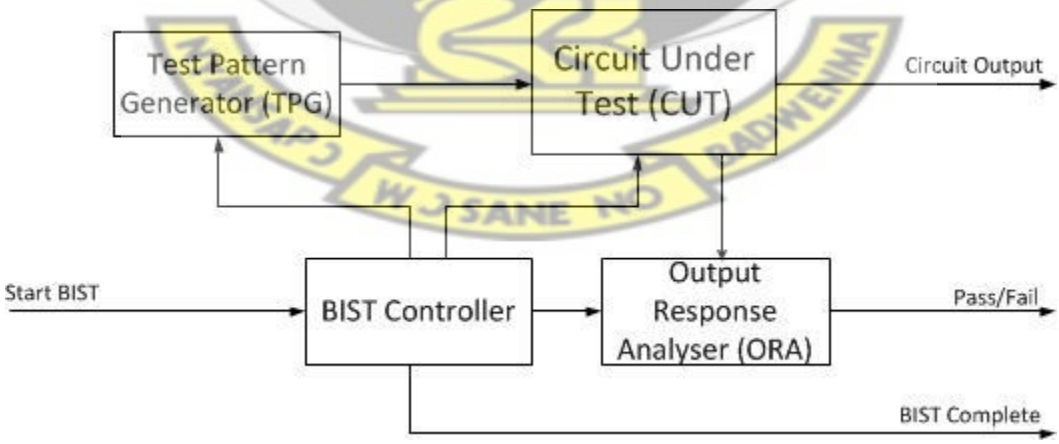


Figure 1.1 - Typical BIST Architecture

1.1 Problem Definition

Considering the many uses of PLLs in modern electronics systems, an integrated PLL with a frequency range practical for simple applications would be an invaluable asset. Indeed, many such PLLs do exist. However, as the size and complexity of VLSI systems, which may be made up of such building blocks as PLLs, continue to increase exponentially, testing of these systems becomes increasingly difficult.

Testing remains a challenge for engineers mainly because of the cost involved in using expensive external automatic test equipment, time, and the difficulty of isolating individual building blocks for testing. One way to solve this problem is the effective implementation of built-in self-test in PLLs.

1.2 Statement of Objectives

This work is aimed at designing a PLL with built-in circuitry to enable self-test. The PLL will be used as a frequency synthesizer and should be fully functional with a practical operating frequency range of 40MHz to 100MHz. The phase noise generated in the PLL should be low enough for simple applications.

The BIST scheme integrated with the PLL should make use of minimal extra hardware and should be easy to implement. It should work well with the designed PLL but must be portable to similar PLL designs. It should also be capable of detecting a large percentage of common faults such as drain-open, source-open, stuck-at-1, and stuch-at-0 faults.

The entire system should be designed in a typical CMOS process using a supply voltage that is common in today's portable products. Finally, simulation of all building blocks as well as system level simulation should be carried out to verify the correct behaviour of the designed system.

1.3 Justification

Almost all physical systems are liable to faults at one point or another in their life cycle. For VLSI systems, faults may develop at any point in the design, fabrication, packaging or even the testing stages. Even if an integrated circuit or perhaps an entire VLSI system is fault-free at the time of production, external factors as temperature and pressure and variations in these thereof could cause such systems to develop faults later on. Regular testing of ICs for correctness and reliability is therefore just as important as the design of the IC itself. Testing becomes necessary to ensure the continuous correct operation of both small and large systems. As previously noted though, traditional methods available for testing VLSI systems are both complicated and expensive. This makes it practically impossible to test poorly funded VLSI projects and systems on a regular basis. Institutions such as schools as well as smaller firms which make use of such systems either have to transport these to specialized companies and pay huge sums or simply replace the entire system. In either case, there is a huge loss. Even for large design and manufacturing companies or well-funded institutions, the described testing process is still an unattractive solution.

Built-In Self-Test offers a practical solution to the integrated circuit testing problem. BIST makes it possible to solve the testing problem by making use of a hierarchical approach. Suppose a complex VLSI system consists of boards which are made up of chips such as PLLs, memory, etc, and that each of the chips has BIST implemented. To test any of the on-board chips the system sends a control signal to the board which in turn activates BIST in the chip. The result is subsequently passed back to the system. Using this approach, built-in test circuitry can be used to test chips, boards and entire systems, while saving cost and time.

This makes it possible for smaller institutions and even individual system owners to test and locate faults. Effective use of BIST should also make remote testing possible. Integrated circuits that are by nature of use, physically difficult to isolate, such as those used in satellites in space or underground systems can still be tested efficiently. Systems in which physical contact is possible but not preferable, such as those used in nuclear reactors, can still be tested using BIST.

1.4 Organisation of Thesis

Chapter one presents a brief introduction to the concept and relevance of phase-locked loops. Problems associated with traditional methods of testing PLLs are presented along with an introduction to how BIST solves these problems. In addition chapter one summarizes the objectives and presents a brief overview of the entire thesis.

Chapter two reviews existing literature on the design of phase-locked loops. The major points of current literature on BIST systems, including their advantages and disadvantages are also presented in this chapter.

The complete design of the PLL frequency synthesizer together with the BIST scheme is presented in detail in chapter 3. How system level parameters of the PLL are determined are explained and the final transistor level schematics of all building blocks are presented. The BIST hardware is also designed and implemented.

Results of simulations of all building blocks and system level designs developed in chapter 3 are presented in chapter 4. Where applicable standard measurements are made and recorded.

Chapter five presents the conclusions of this thesis and makes recommendations for future work. Finally, all schematics exactly as designed in CADENCE IC® are included in the Appendix.

Chapter 2

PLL and BIST Fundamentals

2.0 Introduction

An overview of the operation of the basic phase-locked loop including how it is utilized in frequency synthesis is presented in this chapter. The most common PLL architectures are discussed and compared. The discussion is initially done in relation to the simple PLL which is subsequently compared with the charge pump PLL. All the building blocks that make up a charge pump PLL are then discussed. Finally, the various BIST methods and algorithms available for testing phase-locked loops are studied.

2.1 Basic PLL Operation

A PLL is a circuit that causes a particular system to track another [3]. It keeps an output signal synchronized with a reference input signal in frequency as well as in phase. The PLL may also be described as a servo system which controls the phase of its output signal in such a way that the phase error between output phase and reference phase reduces to a minimum. When the output signal is in sync with the reference clock, at which point the PLL is said to be in a locked state, the phase error between the oscillator's output signal and reference signal is zero, or remains constant.

If a phase error builds up, a control mechanism acts on the oscillator in such a way that the phase error is again reduced to a minimum. In such a control system the phase of the output signal is actually locked to the phase of the reference signal. This is why it is referred to as a *phase-locked loop* [4].

The operating principle of the PLL will be explained by the example of the Linear PLL (LPLL). However, there are other types of PLLs including Digital PLLs (DPLL), All-digital PLLs (ADPLL), and Software PLLs (SPLL).

The PLL consists of three basic functional blocks [2]. They are:

i. A Phase Detector (PD)

ii. A Loop Filter (LF)

iii. A Voltage Controlled Oscillator (VCO)

The first block is the phase detector (PD). The phase detector compares the phase difference between the input reference signal and the oscillator's output signal. The output of the phase detector is a function of the phase difference between the reference and output signal. The main difference between the classic digital PLL and the analog PLL is that the classic digital PLL uses logic gates to realize the phase detector, while the analog PLL uses a multiplier [4]. A traditional analog PLL is shown in Figure 2.1.

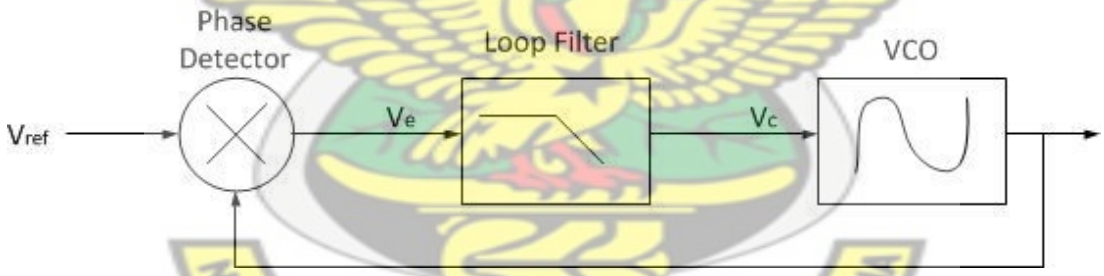


Figure 2.1 - Classic Analog PLL Block Diagram

The phase detector's average output voltage is proportional to the phase difference, $\Delta\phi$, between the two inputs. In the ideal case, the relationship between the average output voltage and the input phase difference is linear, crossing the origin for $\Delta\phi$=0 as shown in Figure 2.2.

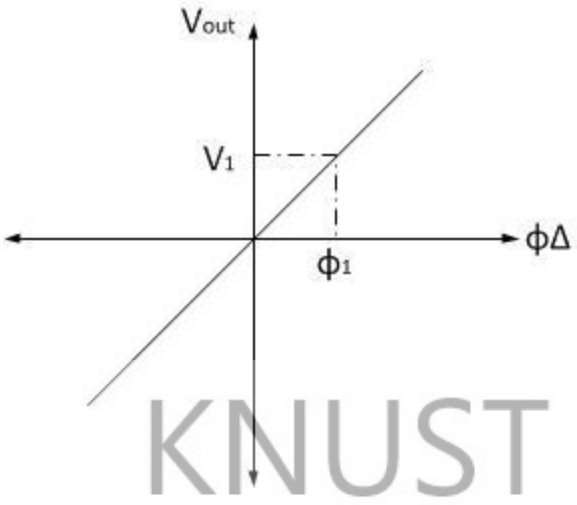


Figure 2.2 - Phase Detector Characteristics

The slope of the line is the gain of the phase detector. It is represented by K_{PD} and is expressed in V/rad. If both the reference and output frequencies are sinusoidal signals, then $V_{ref}(t) = Acos(\omega_r t + \theta_r)$, and $v_{out}(t) = Bcos(\omega_r t + \theta_r)$. Hence the phase detector output which is the sum of of V_{ref} and V_{out} is given by :

$$v_e(t) = \frac{K_m AB}{2}\{cos[(\omega_r - \omega_0)t + \theta_r - \theta_0] + cos[(\omega_r + \omega_0)t + \theta_r + \theta_0]\} \quad \textbf{(2.1)}$$

where K_m is the gain of the multiplier. This phase detector output has a low frequency component that is a function of the phase difference of the two signals and a high frequency component that is a function of the phase summation of the two signals. In the analog PLL, the loop filter is a low pass filter used to filter out the high frequency component of the phase detector output to produce the VCO control voltage, V_c . This high frequency component is a function of the phase summation of the two signals in Equation (2.1. In the classic digital PLL the loop filter averages the phase detector output.

The filtered control voltage is then applied to the input of the third block, which is the voltage-controlled oscillator. The voltage-controlled oscillator produces an output signal, v_{out} , with an angular frequency, ω_{out} , that is controlled by the output voltage of the loop filter, V_c. The

relationship between the angular frequency and the VCO output signal is given by:

$$\omega_{out}(t) = \omega_0 + K_{vco} v_c(t) \tag{2.2}$$

where $\omega_0$is the center frequency of the VCO and K_{vco} is the voltage-controlled oscillator's conversion gain. The output phase is equal to the integral over the frequency variation $\Delta\omega_{out}(t)$.

$$\theta_{out}(t) = \int \Delta\omega_{out}(t)dt = K_{VCO} \int v_c\,(t)dt \tag{2.3}$$

Hence, the control voltage forces the VCO to change the frequency in the direction that reduces the difference between input frequency and output frequency. If the two frequencies are sufficiently close, the PLL feedback mechanism forces the two PD input frequencies to be equal and the VCO is locked with incoming frequency.

2.2 Classification of PLL Types

The very first phase-locked loops were implemented as early as 1932 by de Bellescize [3].The PLL found broader industrial applications only when it became available as an integrated circuit. The first PLL ICs appeared around 1965 and were purely analog devices [13].

An analog multiplier was used as the phase detector, the loop filter was built from a passive or active RC filter, and a voltage-controlled oscillator (VCO) was used to generate the output signal of the PLL. This type of PLL is referred to as the "linear PLL" (LPLL) today. In the following years the PLL drifted slowly but steadily into digital territory. The very first digital PLL (DPLL), which appeared around 1970, was in effect a hybrid device: only the phase detector was built from a digital circuit, but the remaining blocks were still analog [13]. A few years later, the "all-digital" PLL (ADPLL) was invented. The ADPLL is exclusively built from digital function blocks and hence doesn't contain any passive components like resistors and capacitors. A numerically-controlled oscillator is used to generate the output.

Just like filters, PLLs can also be implemented by software. In this case, the function of the PLL is no longer performed by a piece of specialized hardware, but rather by a computer program. This last type of PLL is referred to as an SPLL [4]. The software PLL is normally implemented by a hardware platform such as a microcontroller or a digital signal processor (DSP). The PLL function is realized by software. This offers the greatest flexibility, because a vast number of different algorithms can be developed. For example, an SPLL can be programmed to behave like an LPLL, a DPLL, or an ADPLL.

PLLs may also be classified as 'mixed' signal. The term "mixed" indicates that these PLLs are generally hybrids that contain both linear and digital circuits. Strictly speaking, only the DPLL is a mixed-signal circuit; the LPLL is purely analog. The ADPLL behaves quite differently from mixed-signal PLLs.

PLLs are also classified according to the type of loop filter used in the architecture. The order of the loop filter then determines the type of PLL. If a 1^{st} order loop filter is used, it is referred to as a type I PLL. If a 2^{nd} order filter is used, it is referred to as a type II PLL and so on.

A PLL which uses a phase frequency detector instead of a normal phase detector, accompanied with a charge pump is called a Charge Pump PLL. A PLL with only a phase detector and no charge pump is also sometimes referred to as a Simple PLL.

2.3 Common PLL Terminology

A few of the common parameters associated with PLLs will be encountered in the linear analysis in section 2.4. These are explained here.

2.3.1 PLL Bandwidth

The bandwidth is the frequency at which the PLL starts to move out of phase and frequency lock in relation to the reference signal [14]. It provides a measure by which PLL's ability to track the input clock can be determined. It is commonly designated by the symbol ω_{3db}.

2.3.2 Natural Frequency

The natural frequency ω_n, is derived from the bandwidth and serves as a measure of the response time of the PLL [14]. It is preferable that the natural frequency be as high as possible as this will decrease the PLL lock time.

2.3.3 Damping Factor

The damping factor of the PLL, commonly designated ζ, is a measure of the overshoot and ringing of the PLL frequency response [14]. Ideally, the damping factor should be near 0.707 at which point critical damping occurs.

2.4 PLL Linear Analysis

The PLL is a highly non-linear system and its mathematical analysis is quite difficult [15]. To facilitate the analysis, the conventional theory neglects the VCO dynamics and ripple produced by the phase detector. By neglecting these, the amount of information about the actual PLL performance the mathematical analysis can extract is significantly reduced [16]. Nevertheless, this usage of a linear theory is not a serious problem because quite a number of PLL applications are related to high frequencies with a small variation [16]. For these applications, the needs of engineering practice are satisfied by the conventional small signal analysis.

The PLL is best described with a linear model if the loop is in lock. The loop is in lock when the phase error signal produced by the phase detector settles on a constant value. This implies that the output signal has the same frequency as the input reference signal. A phase difference between the reference and output signal may still exist depending on the type of PLL that is used. However, this phase difference remains constant while the loop is in lock. If the PLL is used as a frequency synthesizer, the output signal will have a frequency N times the reference frequency.

2.4.1 Mathematical Model for the locked state

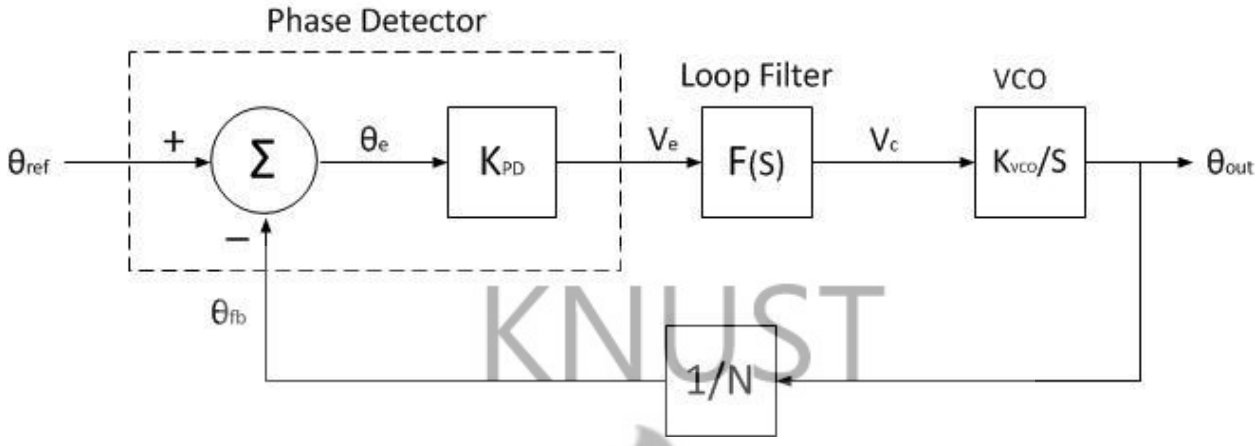


Figure 2.3 - Linear control model of a PLL in locked state

The analysis presented in this section is a simplified form of that appearing in [17]. The building blocks of Figure 2.3 are taken as a basis for the mathematical model of a PLL in lock. Since a loop division factor N is included in this model, the same mathematical model developed is applicable to any frequency synthesizer. N can be considered equal to one for PLLs with no loop dividers.

The following analysis shows step by step how to obtain the PLL transfer function. From first principles, the transfer function can simply be quoted as the output phase divided by the input or reference phase. This is shown below:

$$H(s) = \frac{\theta_{out}}{\theta_{ref}} \tag{2.4}$$

The phase detector sums the input reference phase, θ_{ref}, with the phase of the feedback signal, θ_{fb}, and amplifies the difference with a gain K_{PD} to produce an error voltage, $V_e(s)$. This error voltage can be computed as:

$$V_e(s) = K_{PD}\left[\theta_{ref}(s) - \theta_{out}(s)\right] = K_{PD}\theta_e(s) \tag{2.5}$$

This error voltage is filtered by the loop filter to produce the VCO control voltage that is equal to the following:

$$V_C(s) = V_e(s)F(s) \tag{2.6}$$

As shown by Equation (2.3, the VCO can be modeled as a phase integrator. This results in an output phase, θ_{out}, equal to the following:

$$\theta_{out}(s) = \frac{V_c(s)K_{VCO}}{s} \tag{2.7}$$

The output phase is fed back and passes through a loop divider where it is divided by a factor of *N* to generate the feedback phase, θ_{fb}, equal to the following:

$$\theta_{fb}(s) = \frac{\theta_{out}(s)}{N} \tag{2.8}$$

2.4.2 Definition of Transfer Functions

Based on all the above equations, the transfer function of the PLL, $H(s)$, is equal to the following:

$$H(s) = \frac{K_{PD}K_{VCO}F(s)}{s + \frac{K_{PD}K_{VCO}F(s)}{N}} \tag{2.9}$$

Also the phase error transfer function is equal to the following:

$$\frac{\theta_e(s)}{\theta_{ref}(s)} = \frac{s}{s + \frac{K_{PD}K_{VCO}F(s)}{N}} \tag{2.10}$$

The VCO control voltage transfer function is also given by the following:

$$\frac{V_c(s)}{\theta_{ref}(s)} = \frac{sK_{PD}F(s)}{s + \frac{K_{PD}K_{VCO}F(s)}{N}} \tag{2.11}$$

Some basic deductions can be made from the three transfer functions obtained in Equations (2.9, (2.10, and (2.11.

The PLL transfer function given in equation (2.9, has a low-pass characteristic with a gain of *N*. This means that for slow (low frequency) variations in the reference phase, the loop will basically track the input signal and produce an output phase that is N times larger. Thus the output frequency is N times the input reference frequency. The phase error transfer function, given in equation (2.10, has a high-pass characteristic. This implies that for slow variations in the reference phase, the phase error will be small. However, fast (high frequency) variations in the reference phase will not be filtered and show up as a phase error. The VCO control voltage transfer function can be viewed as the filtered phase error output. It also has a high-pass characteristic. However, depending on the parameters of the loop filter, it can take on a more band-pass shape.

Generally, the dynamics of the PLL are dependent on the type of loop filter used. The passive lag filter shown in Figure 2.4 is a common filter used in PLL design [4]. It is now employed in developing further the mathematical model of the PLL.

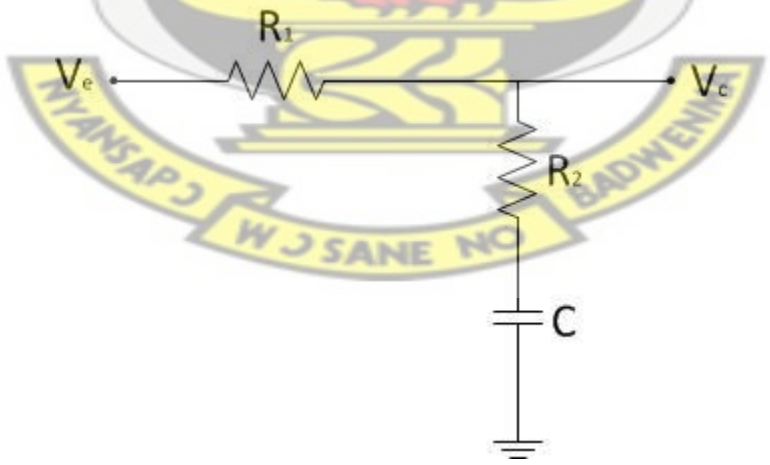


Figure 2.4 - Passive Lag Filter

This filter is quite straightforward to build and proves to be adequate in most applications. If filter gain is necessary for increased tracking accuracy, active filters with a high gain opamp may

be used. The transfer function of the passive lag filter is given by:

$$F(s) = \frac{V_c(s)}{V_e(s)} = \frac{sCR_2 + 1}{sC(R_1 + R_2) + 1} \quad \textbf{(2.12)}$$

Substituting equation (2.12 into the PLL transfer equation obtained in equation (2.9, the following equation is obtained.

$$H(s) = \frac{K_{PD}K_{VCO}\left(\frac{sCR_2 + 1}{C(R_1 + R_2)}\right)}{s^2 + s\frac{K_{PD}K_{VCO}CR_2 + N}{NC(R_1 + R_2)} + \frac{K_{PD}K_{VCO}}{NC(R_1 + R_2)}} \quad \textbf{(2.13)}$$

It can be observed that using a first order filter in the PLL results in a second order system. In fact, the order of a PLL is equal to the loop filter order plus one. The second order PLL system can be described in a standard control system format as follows:

$$H(s) = N\frac{s\left(2\zeta\omega_n - \frac{N\omega_n^2}{K_{PD}K_{VCO}}\right) + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad \textbf{(2.14)}$$

where ζ is given by:

$$\zeta = \frac{1}{2}\sqrt{\frac{K_{PD}K_{VCO}}{NC(R_1 + R_2)}}\left(CR_2 + \frac{N}{K_{PD}K_{VCO}}\right) \quad \textbf{(2.15)}$$

and ω_n is also given by:

$$\omega_n = \sqrt{\frac{K_{PD}K_{VCO}}{NC(R_1 + R_2)}} \quad \textbf{(2.16)}$$

2.4.3 Effects of the damping factor and natural frequency

The parameter ζ is the damping factor and ω_n is the natural frequency. The s-plane plot shown in Figure 2.5 can be used to explain these parameters [18].

It can observe that the poles are located at a distance ω_n from the origin and at an angle $\theta = sin^{-1}\zeta$. The damping factor, ζ , is a measure of stability. If ζ is equal to zero, then the poles of the system lie on the imaginary axis at a distance ω_n from the origin. For this case the impulse response of the system results in a steady oscillation at a frequency ω_n. On the other hand, as ζ is increased, the poles move to the left-hand plane and the system becomes stable.
In this specific situation, the impulse response of the system becomes a damped oscillation at a frequency ω_n.

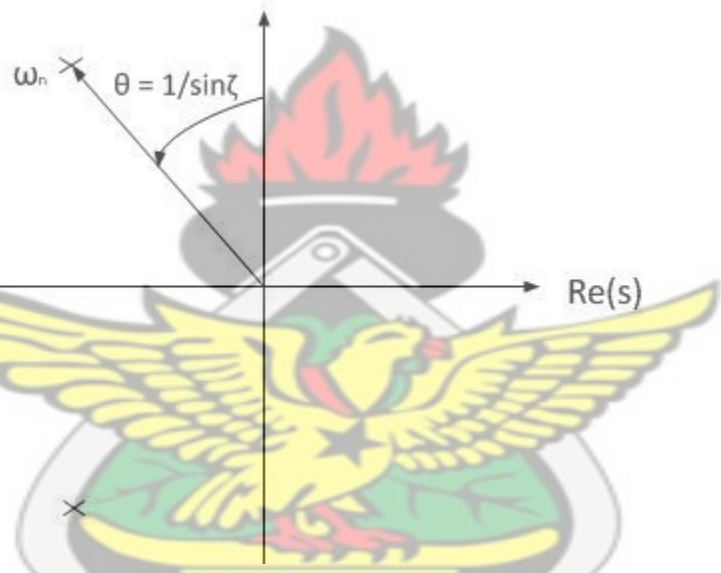


Figure 2.5 - S-Plane plot showing complex poles of the natural frequency

The PLL transfer function, written in the standard control system format, as shown in equation (2.14 can be used to plot the PLL's frequency response for different damping factors. This plot is shown in Figure 2.6.

The PLL frequency response shows the expected second-order low-pass characteristic. The Q value of a PLL is inversely proportional to the damping factor, ζ . High Q values display a frequency response with a sharp peak at ω_n. This results in an oscillatory transient response. If the damping factor is high, the Q value of the system is low and the frequency response is flat

across a wide bandwidth. This results in a slow, sluggish transient response.

Generally, an optimally flat frequency transfer function is desired [4]. This occurs when $\zeta = {}^{1}/_{\sqrt{2}} \approx 0.707$, which corresponds to a second-order Butterworth low-pass filter.

As in amplifiers, the bandwidth of a PLL is often specified by the 3-dB corner frequency ω_{3db}.

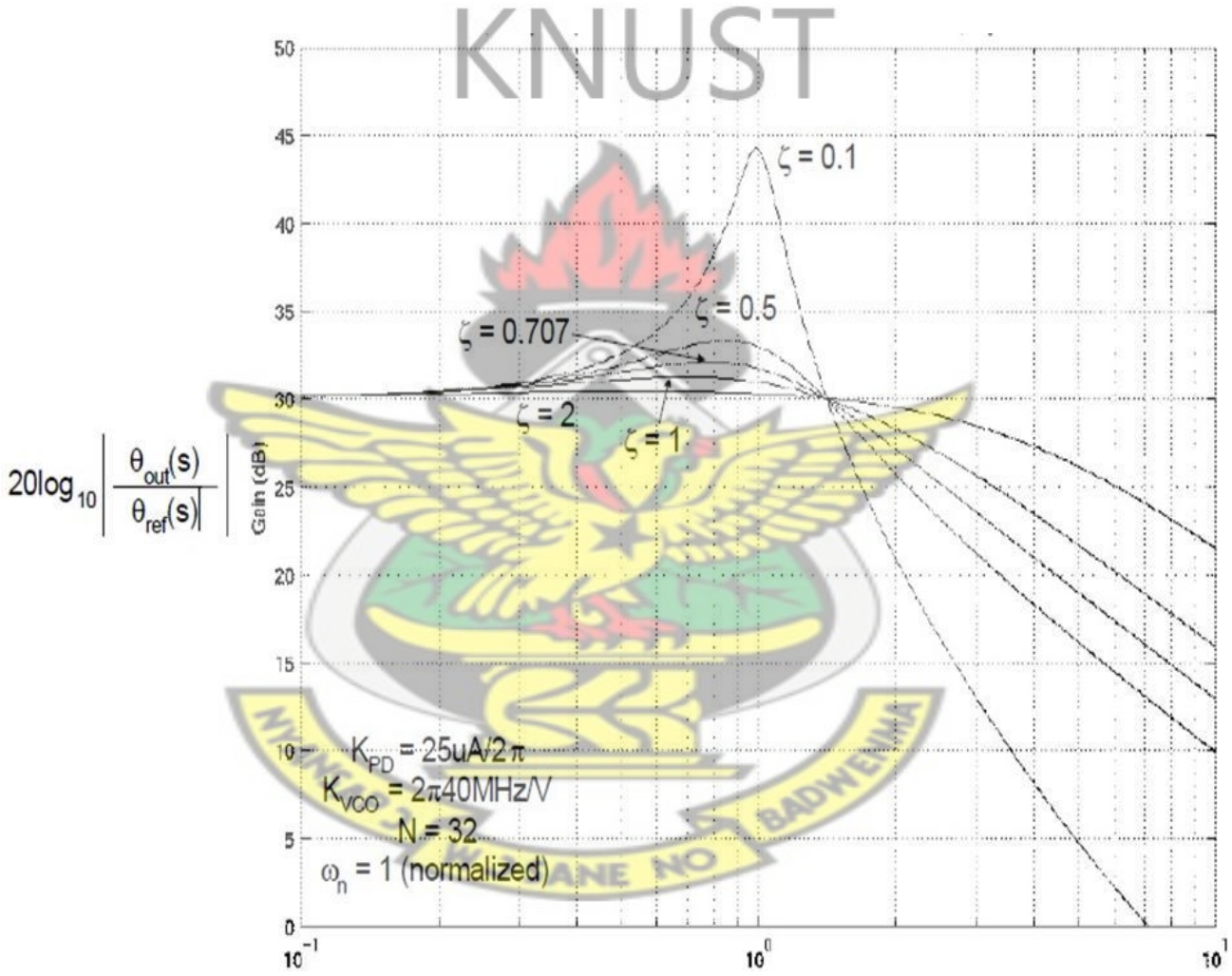


Figure 2.6 - Frequency response of 2nd order PLL with different damping factors

A high bandwidth PLL provides a fast lock time and tracks jitter on the reference clock source, passing it through to the PLL output. A low bandwidth PLL filters out reference clock jitter, but increases lock time. The 3-db bandwidth of the PLL is given by [4]:

$$\omega_{3db} = \omega_n \left[1 + 2\zeta^2 + \sqrt{(1 + 2\zeta^2)^2 + 1}\right]^{1/2} \quad \textbf{(2.17)}$$

It is clearly seen from this equation that the values of ζ and ω_n have an effect on the bandwidth of the PLL. For a damping factor of $\zeta = 0.7$, $\omega_{db} = 2.06\omega_n$, which is about twice the natural frequency.

2.5 Frequency Synthesizers

Phase-locked loops, as mentioned earlier, have a variety of applications. However, in this section, attention is given to their use as frequency synthesizers. A frequency synthesizer (FS) is a device capable of generating a set of signals of given output frequencies with very high accuracy and precision from a single reference frequency. The signal generated at the output of the frequency synthesizer is commonly known as a local oscillator signal, since it is used in communication systems as the reference oscillator for frequency translation [2].

PLL frequency synthesizers are widely used in all forms of radio communications equipment today. From cellular phones to various wireless products and domestic radios and televisions to professional radio communications equipment, the local oscillators virtually all use PLL frequency synthesizers. These offer many advantages over the use of other forms of local oscillators. Frequency synthesizers not only offer high levels of stability and accuracy, they are also easy to control from digital circuitry such as microprocessors [13]. This enables facilities such as keypad frequency entry, channel memories and more to be implemented. In view of all their advantages, PLL frequency synthesizers are usually the preferred form of local oscillator for most applications. A phase locked loop needs some additional circuitry if it is to be converted into a frequency synthesizer. This is done by adding a frequency divider or loop divider between the voltage controlled oscillator and the phase comparator. Figure 2.7 shows a representation of this system.

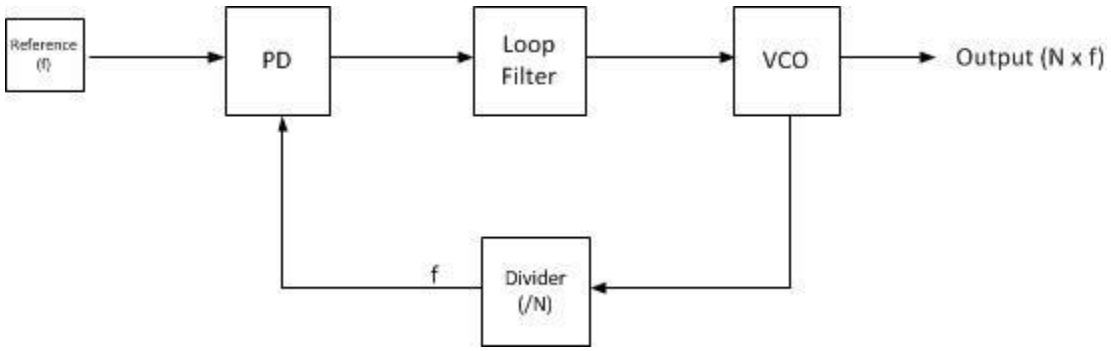


Figure 2.7 - A PLL with a divider - a frequency synthesizer

A reference oscillator is included in the block diagram. Although this is not strictly part of the loop itself, the reference signal is required for the synthesizer's operation. When the divider is added into the circuit, the PLL still tries to reduce the phase difference between the two signals entering the phase comparator. Again when the circuit is in lock both signals entering the comparator are exactly the same in frequency. For this to be true the voltage controlled oscillator must be running at a frequency equal to the phase comparison frequency times the division ratio.

It can be seen that if the division ratio is altered by one, then the voltage controlled oscillator will have to change to the next multiple of the reference frequency. This means that the step frequency of the synthesizer is equal to the frequency entering the comparator. Most synthesizers need to be able to step in much smaller increments if they are to be of any use [19]. This means that the comparison frequency must be reduced. This is usually achieved by the use of a 1 MHz or so reference oscillator, and then dividing this signal down to the required frequency using a fixed divider. In this way a low comparison frequency can be achieved.

Depending on the nature of the digital divider used, a PLL frequency synthesizer may be classified as either an Integer-N synthesizer or Fractional-N synthesizer.

2.5.1.1 Integer-N Frequency Synthesizers

This is a system designed to create a set of frequencies that are an integer multiple of a fixed

reference frequency [13]. Hence the division factor is an integer. In the circuit of Figure 2.8, it is shown that this scaling factor *N* may not be a fixed value but rather set by an external digital signal [4].

In most of the frequency-synthesizer ICs presently available, a reference divider is integrated on the chip. CMOS is the preferred technology today because of its low power consumption, high noise immunity, and large range of supply voltages, [20]. However, the limited speed

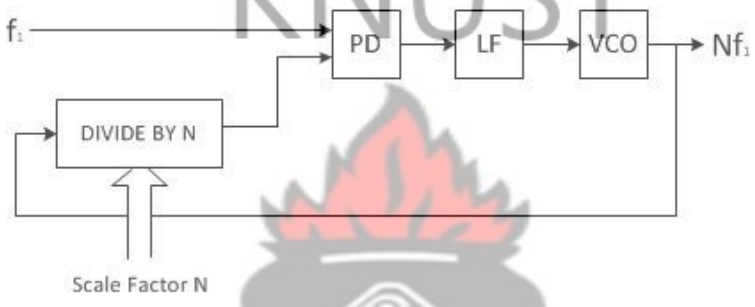


Figure 2.8 - Frequency synthesizer block diagram

of CMOS devices precludes their application for directly generating frequencies in the range of 100MHz or more.

To generate higher frequencies in synthesizers, prescalers are used [1]. These prescalers are built with other IC technologies such as TTL to avoid the limitations of CMOS. Such prescalers extend the range of frequencies into the microwave frequency bands. The use of a simple prescaler is depicted in Figure 2.9.

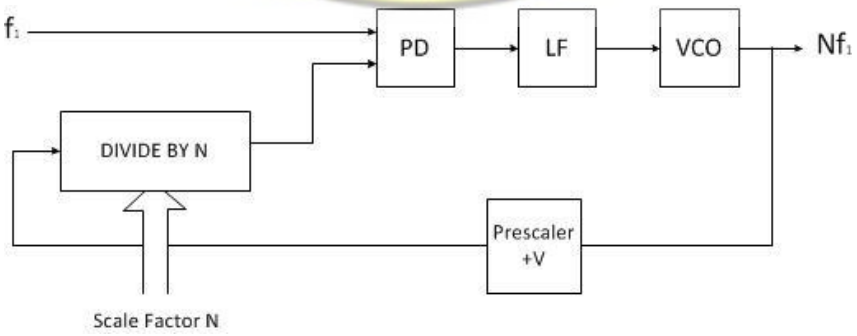


Figure 2.9 - PLL frequency synthesizer with prescaler

If the scaling factor of the prescaler is V, the output frequency of the synthesizer becomes

$$f_{out} = NVf_1 \quad \textbf{(2.18)}$$

Using a scaling factor V, which is equal to one, defeats the entire purpose of using a prescaler in the first place. V should therefore be much greater than 1 in most cases. A consequence of this however, is that it is no longer possible to generate every desired integer multiple of the reference frequency f_1. As an example, if *V* is 10, only output frequencies of $10f_1, 20f_1, 30f_1$,... can be generated. This disadvantage can be circumvented by using a so-called dual-modulus prescaler.

A dual-modulus prescaler is simply a counter whose division ratio can be switched from one value to another by an external control signal [6]. The use of the dual-modulus prescaler makes it possible to generate a number of output frequencies that are spaced only by f_1 and not by a multiple of f_1.

2.5.1.2 Fractional-N Frequency Synthesizers

In contrast to the integer-N frequency synthesizer, this device is able to create frequencies that are *N.f* times a reference frequency, where *N* is the integer part and *f* is the fractional part of an arbitrary number [21].

A modification to the basic Integer-N synthesizer which is shown in Figure 2.10 can be obtained by inserting an additional divide by *m* counter outside the loop. The result is a very simple fractional-N synthesizer.

By adjusting N and M you can obtain a fractional rate frequency multiplier suitable for frequency synthesis. The output frequency is given by:

$$f_{out} = \frac{N}{M} f_{in} \quad \textbf{(2.19)}$$

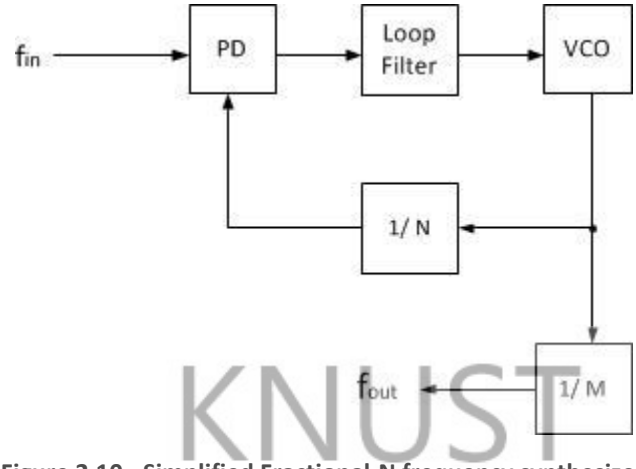


Figure 2.10 - Simplified Fractional-N frequency synthesizer

2.5.2 Analogue PLL frequency synthesizers

Placing a digital divider is not the only method of making a synthesizer using a phase locked loop. It is also possible to use a mixer in the loop (Figure 2.11). Using this technique places an offset into the frequency generated by the loop [22]. The PLL's operation with the mixer incorporated can be analyzed in the same manner as the loop with a divider. When the loop is in lock the signals entering the phase detector are at exactly the same frequencies. The mixer adds an offset equal to the frequency of the signal entering the other port of the mixer.

To illustrate the way this operates values have been included in the block diagram of Figure 2.11. If the reference oscillator is operating at a frequency of 10 MHz and the external signal is at 15 MHz then the VCO must operate at either 5 MHz or 25 MHz. Normally the loop is set up so that mixer adds up to the frequency, and if this is the case then the oscillator will be operating at 25 MHz.

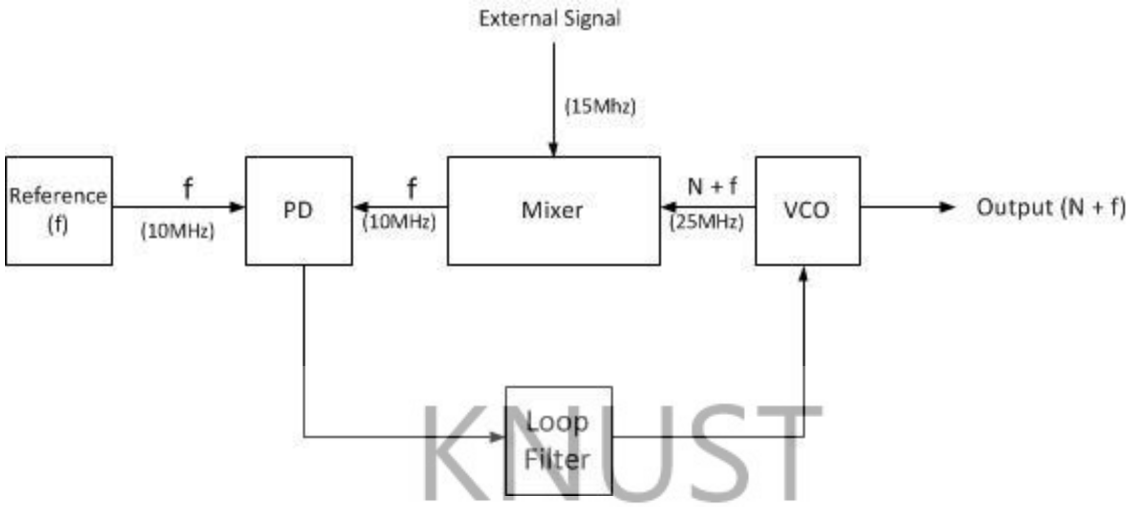


Figure 2.11 - Analog PLL frequency synthesizer

2.6 Charge Pump Phase-locked Loops

For type I PLLs, there are always trade-offs between the loop filter damping ratio, the loop filter bandwidth and the phase error. Hence the performance of the PLL cannot improve beyond a certain limit. Apart from this, a simple PLL suffers from the critical drawback of limited acquisition range [20].

Suppose when a PLL circuit is turned on, its oscillator operates at a frequency far from the input frequency, i.e., the loop is not locked. Now the PLL starts acquiring a lock. The transition of the loop from unlocked to locked condition is a very nonlinear process because the phase detector senses unequal frequency. Also for this kind of PLL, the "acquisition range" is on the order of ω_{LPF}, that is, the loop locks only if the difference between ω_{in} and ω_{out} is less than roughly ω_{LPF}. If ω_{LPF} is reduced to suppress the ripple on the control voltage, the acquisition range decreases. However, even if the input frequency has a precisely controlled value, a wide acquisition range is often necessary because the VCO frequency may vary considerably with external factors like temperature.

In order to remove this problem, in the charge pump PLL, frequency detection is also

incorporated in addition to phase detection. The two frequencies (reference and VCO output frequency) are compared until they are equal. Once these two frequencies are equal, phases are compared and the VCO is tuned such that the phases of the reference and feedback waveform are equal. Frequencies are compared using a frequency detector which generates a dc voltage equal to the difference of the two input frequencies and drives the VCO such that ω_{in}=ω_{out}. When $|\omega_{in}$-$\omega_{out}|$ is sufficiently small, the PLL continues in 'normal' operation, acquiring lock. This scheme increases the acquisition range to the tuning range of VCO.

The individual building blocks of the charge pump phase-locked loop will now be discussed in detail.

2.6.1 Phase Frequency Detector

The phase detector for a PLL measures the phase difference between the input signal and the PLL's voltage controlled oscillator. This phase difference is converted to a voltage in the phase detector, which is then used to provide feedback control to the local oscillator [19]. In an ideal detector, the average output, $\overline{V_{out}}$, is linearly proportional to the phase difference, $\Delta\Phi$, between its two inputs as shown in Figure 2.12 [20].

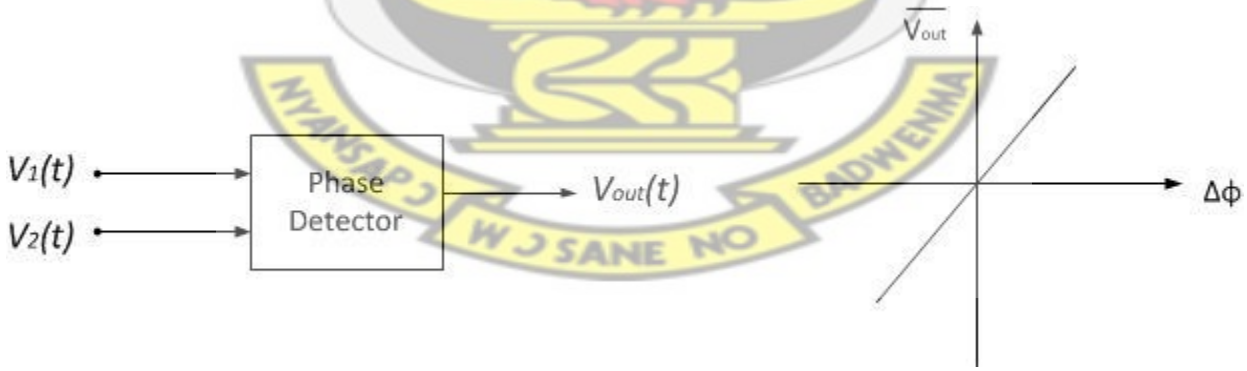


Figure 2.12 - Definition of ideal phase detector

In this ideal case, the relationship between $\overline{V_{out}}$ and $\Delta\Phi$ is linear, crossing the origin at $\Delta\Phi = 0$. The slope of the line K_{PD} expressed in V/rad is the gain of the phase detector. The simplest

example of a traditional phase detector is the exclusive OR (XOR) gate. As shown in Figure 2.13, as the phase difference between the inputs varies, so does the width of the output pulses, thereby providing a dc level proportional to $\Delta\Phi$. Other simple systems such as the JK-flipflop can be used as a phase detector [4].

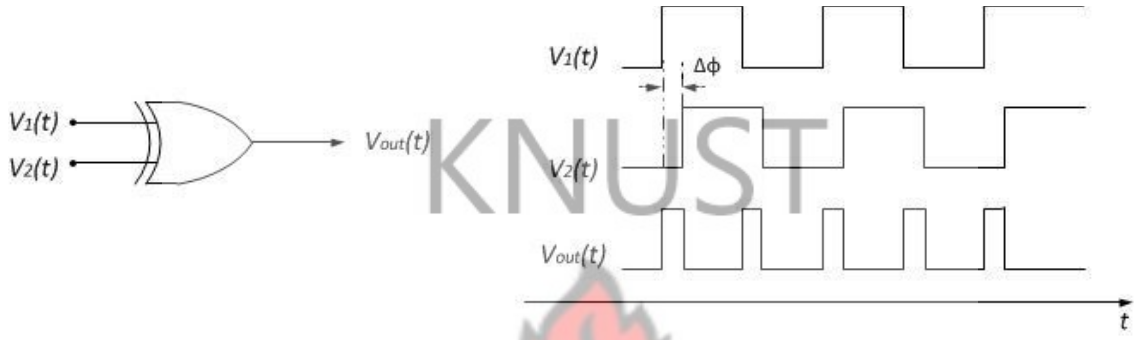


Figure 2.13 - XOR gate as a phase detector

The simple system described above is capable of detecting only differences in phase. However, it is possible to devise a circuit capable of detecting both phase and frequency differences. This is called a phase/frequency detector (PFD) and it is illustrated conceptually in Figure 2.14 [20]. The use of this kind of phase/frequency detector allows for a wide frequency locking range, potentially the entire VCO tuning range.

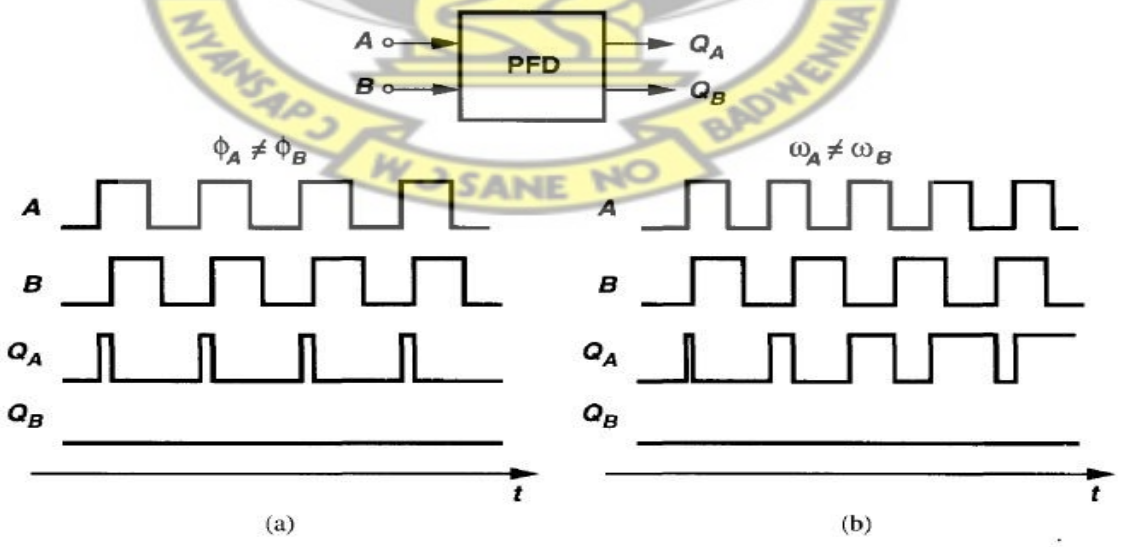


Figure 2.14 - Conceptual operation of a phase/frequency detector

The circuit employs sequential logic to create three states and respond to the rising (or falling) edges of the two inputs. If initially $Q_A = Q_B = 0$, then a rising transition on *A* leads to $Q_A = 1, Q_B = 0$. The circuit remains in this state until *B* goes high, at which point Q_A returns to zero. The behavior is similar for the *B* input.

In Figure 2.14(a), the two inputs have equal frequencies but *A* leads *B.* The output Q_A continues to produce pulses whose width is proportional to $\phi_A - \phi_B$ while Q_B remains at zero. In Figure 2.14(b), *A* has a higher frequency than B and Q_A generates pulses while Q_B does not. By symmetry, if *A* lags *B* or has a lower frequency than *B*, then Q_B produces pulses and Q_A remains quiet. Thus, the dc contents of Q_A and Q_B provide information about $\phi_A - \phi_B$ or $\omega_A - \omega_B$. The outputs Q_A and Q_B are called the "UP" and "DOWN" pulses, respectively. A common implementation of this phase/frequency detector is the architecture shown in Figure 2.15.

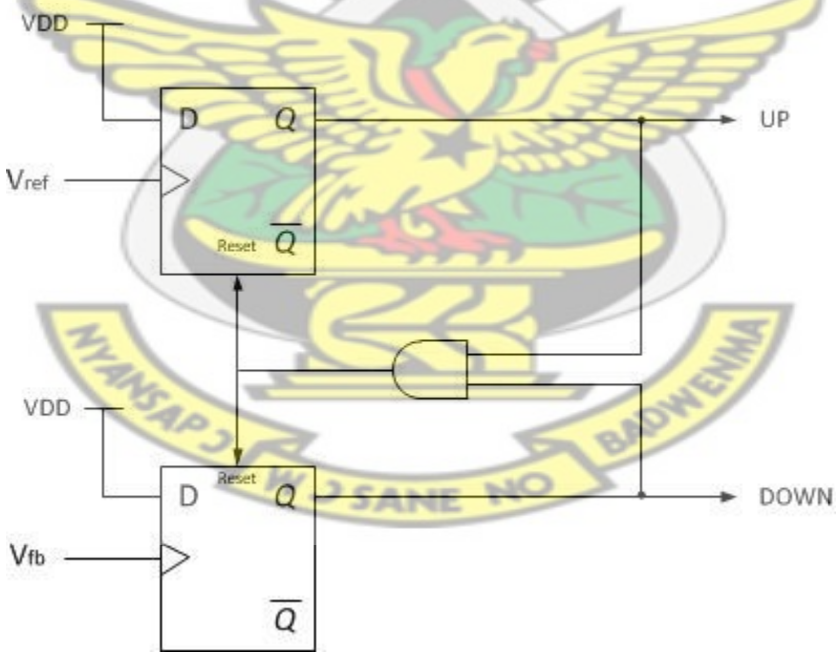


Figure 2.15 - Phase/frequency detector implementation

As stated, the phase/frequency detector produces two output signals, UP and DOWN , that are dependent on the phase and frequency relationship of the two inputs, V_{ref} and V_{fb} . The UP

and DOWN outputs control the charge pump which acts as the phase frequency detector's output stage.

The UP output signal of the PFD goes high on the rising edge of V_{ref}. The DOWN output signal goes high on the rising edge of V_{fb}. The UP and DOWN signals remain high until they are reset by the AND combination of UP and DOWN. In other words, the reset signal is produced when both V_{ref} and V_{fb} clock inputs are high. Both *Q* outputs will be essentially low when both signals are in phase and of the same frequency.

Figure 2.16 [20] shows one possible architecture that could be used to implement the D-flipflop.

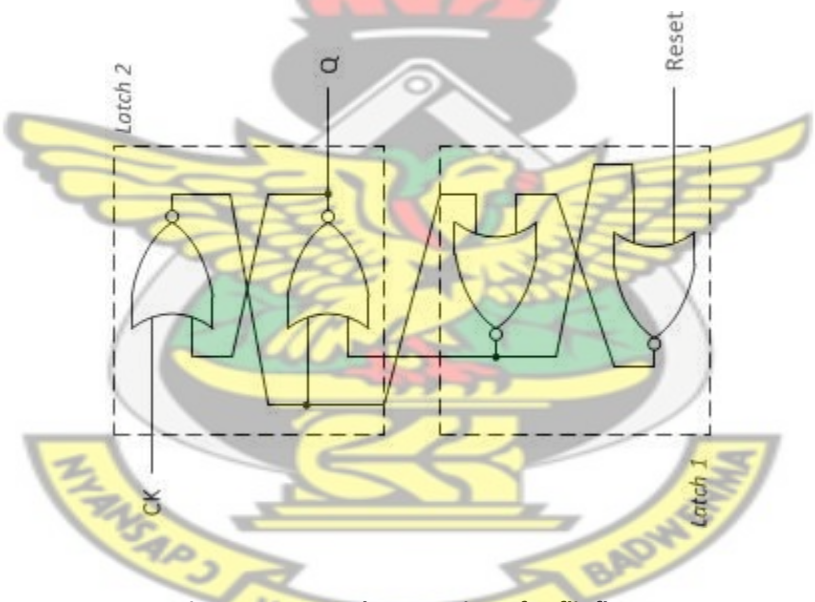


Figure 2.16 - Implementation of D-flipflop

One potential problem that this phase detector may have is a dead zone [23]. The dead zone occurs when the rising edges of the input reference and VCO feedback signals are almost aligned. If the delay through the reset path is shorter than the delay to the charge pump that the PFD is driving then the charge pump will not get switched even though there is a phase error present. The PFD must be designed in such a way as to ensure that the delay through the reset path is longer than the delay to the charge pump.

2.6.2 Charge Pump

A charge pump is a kind of DC convertor that uses a capacitor as an energy storage element. Signals coming from PFD are applied to the charge pump to steer the current into and out of the capacitor causing voltage to increase or decrease accordingly [1]. The capacitor actually forms part of the loop filter and not the charge pump itself. Figure 2.17 shows a basic charge pump circuit and one in operation.

The charge pump injects, subtracts, or leaves alone the charge stored across a capacitor in the low-pass filter, depending on the output of a sequential phase-detector circuit. When S_1 is closed, I flows into the low-pass filter, increasing the control voltage to the VCO. When S_2 is closed, I flows out of the low-pass filter, decreasing the control voltage to the VCO. As illustrated in Figure 2.17(b), this process may be referred to as charging and discharging of the loop filter. When both switches are open, the top plate of the loop filter's first capacitor is open circuited and the output voltage remains constant in the steady state.

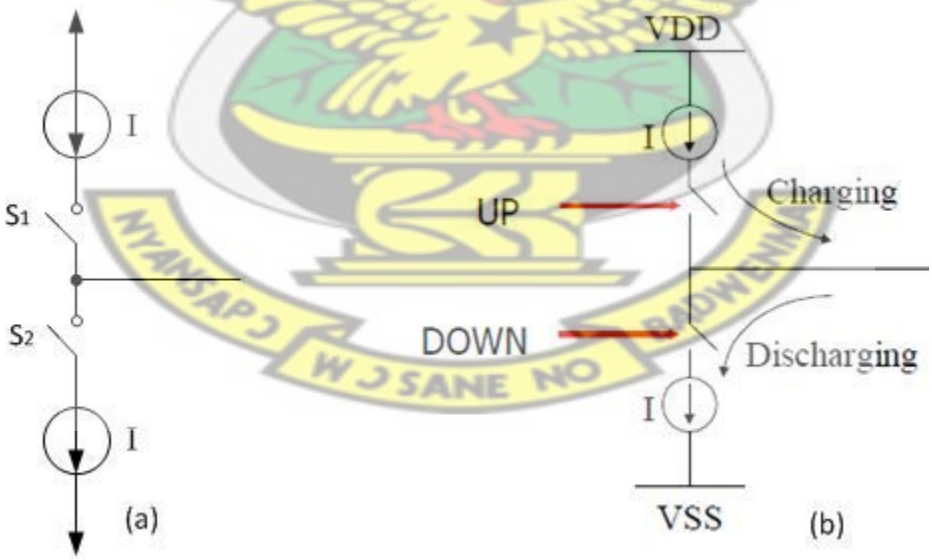


Figure 2.17 - Simple charge pump

2.6.3 Loop Filter

Figure 2.18 shows the loop filter commonly used in charge pump PLLs.

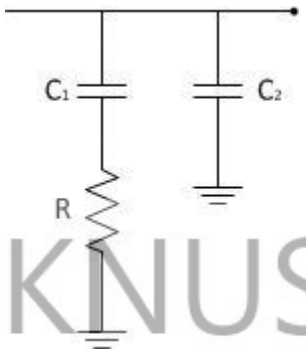


Figure 2.18 - Loop filter

The PLL filter is needed to remove any unwanted high frequency components which might pass out of the phase detector and appear in the VCO tune line. They would then appear on the output of the Voltage Controlled Oscillator, VCO, as spurious signals. To show how this happens take the case when a mixer is used as a phase detector. When the loop is in lock the mixer will produce two signals: the sum and difference frequencies. If the two signals entering the phase detector have the same frequency, the difference frequency is zero and a DC voltage is produced proportional to the phase difference as expected. The sum frequency is also produced and this will fall at a point equal to twice the frequency of the reference. If this signal is not attenuated it will reach the control voltage input to the VCO and give rise to spurious signals. The loop filter takes care of this attenuation.

The resistor, R, is included to realize a zero in the low-pass filter's transfer function. This allows the designer to control the damping of the loop's transfer function separately from the time constant of the loop [23]. The capacitor C_2 is much smaller compared to C_1. It is included to suppress glitches. It has little effect on loop dynamics and may be ignored when developing a simple mathematical model for the filter's behavior.

Ignoring C_2, the loop filter has the following transfer function:

$$F(s) = \frac{V_c(s)}{I_e(s)} = \frac{R\left(s + \frac{1}{RC_1}\right)}{s} \qquad \textbf{2.20}$$

Figure 2.19 shows a plot of this frequency response. It is seen that the pole at zero causes the transfer function to level off at high frequencies as needed.

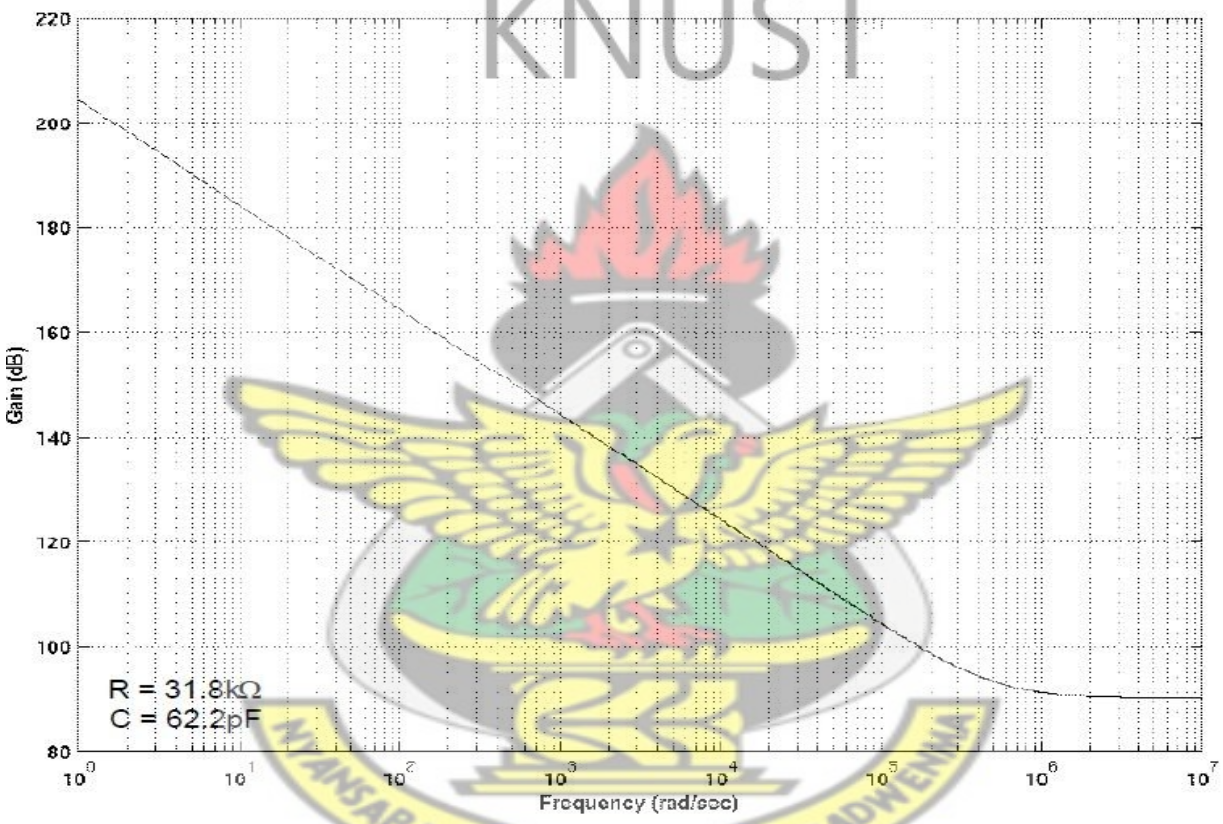


Figure 2.19 - Frequency response of the loop filter

When the secondary capacitor, C_2, is included however, the following transfer function is obtained:

$$F(s) = \frac{\frac{1}{C_2}\left(s + \frac{1}{RC_1}\right)}{s^2 + \frac{s(C_1 + C_2)}{RC_1C_2}} \qquad \textbf{2.21}$$

2.6.4 Voltage Controlled Oscillator

Within a phase locked loop, or frequency synthesizer, the performance of the voltage controlled oscillator is of paramount importance. This is because the VCO performance determines many of the overall performance characteristics of the synthesizer. Generally, a VCO can be treated as a black box with an input V_{tune} and a periodic oscillating output $V(t)$ depicted as drawn in Figure 2.20 [24].

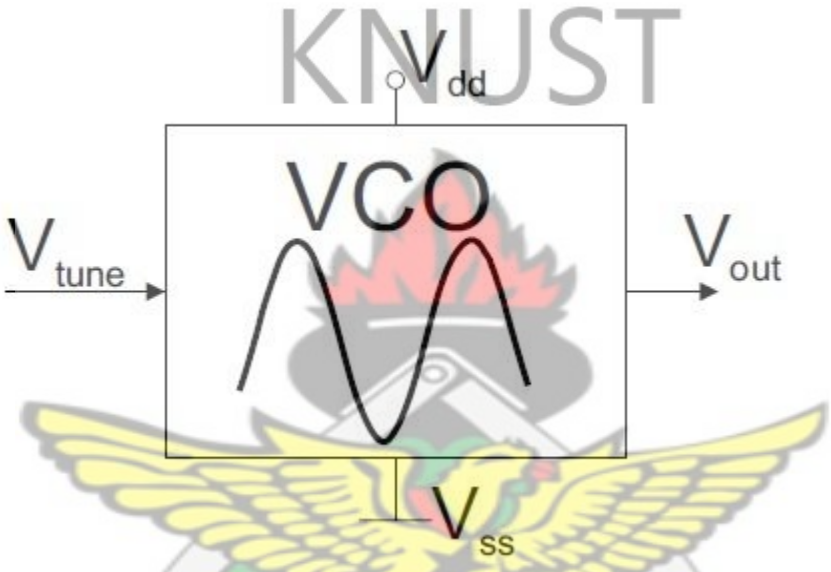


Figure 2.20 - Block level view of VCO

The VCO is connected to the power supply through VSS and VDD. The output voltage $V_{out}(t)$, may be differential or single ended. However, in both cases it is a periodic signal given by:

$$V_{out}(t) = V_o sin(\omega_o + \Phi) \quad \textbf{2.22}$$

Where ϕ is the phase, V_o the amplitude, and ω_o is the angular frequency which is largely dependent on v_{tune} and is given by:

$$\omega_o(v_{tune}) = 2\pi f_o(v_{tune}) \quad \textbf{2.23}$$

The output signal angular frequency can also be written in terms of the control voltage, v_{c}, (or v_{tune}) directly [25]. The relation is shown in equation 2.24

$$\omega_{out} = \omega_o + K_{VCO} v_c \quad \text{2.24}$$

The general transfer characteristic of the VCO is given in Figure 2.21. It has a center frequency of ω_o. The slope of the transfer characteristic in the linear region is equal to the VCO conversion gain which is usually represented by K_{VCO} [17].

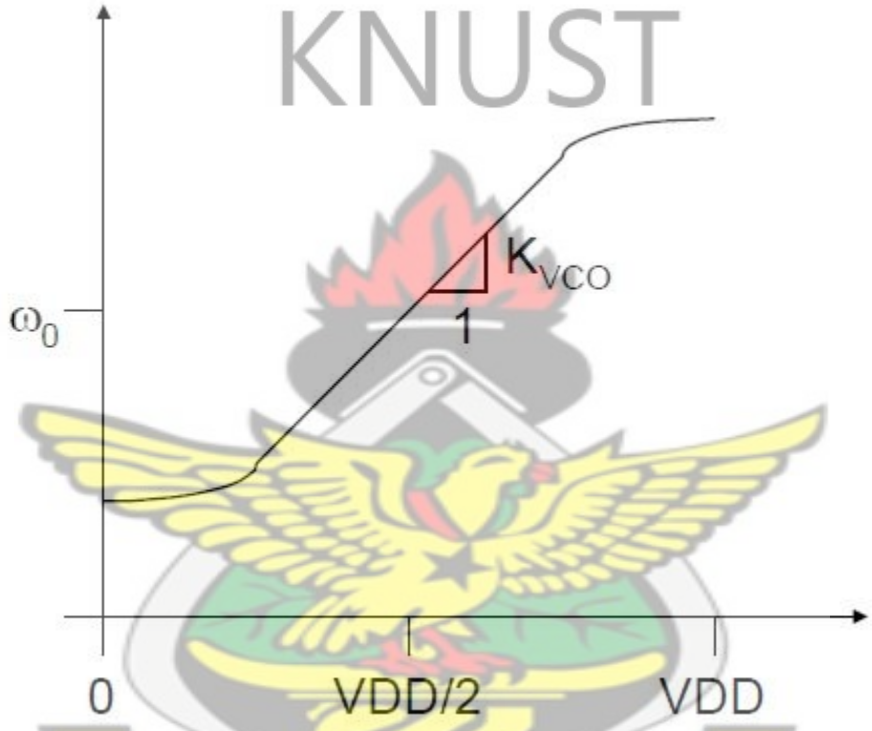


Figure 2.21 - Transfer characteristic curve of VCO

There are many different types of VCOs. However, the most commonly used types of oscillators for PLL design are the ring oscillator and LC oscillator [25], [23].

2.6.4.1 Ring Oscillators

A classical ring oscillator circuit solution is the connection of amplifiers or inverters. If the phase shift over the ring is 360^o, it will oscillate [15]. Practically, for single ended inverters, this will require the number of inverting stages to be odd. An example of this arrangement is shown in Figure 2.22.

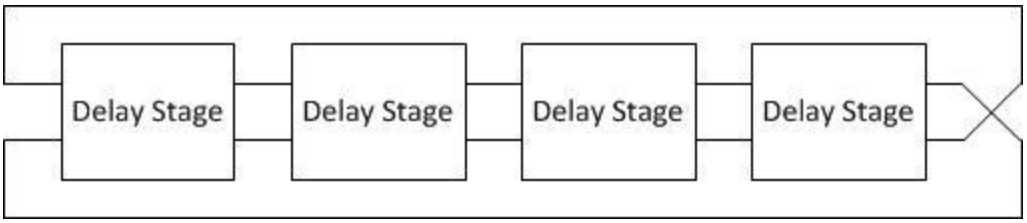


Figure 2.22 - Exemplary ring oscillator topology

Ring oscillators have the advantage of being easier to design and integrate. They also cover a very small area and feature wide tuning ranges. However they suffer from higher power consumption and phase noise [24]. Past work done in ring oscillator design, such as in [26], has shown that it is possible to design low phase noise ring-oscillators. This is still at a much higher power consumption than for LC-VCOs though.

Figure 2.24 shows examples of some ring oscillator topologies [27], [20].

2.6.4.2 LC Oscillators

A general LC-VCO can be symbolized as in Figure 2.23. The oscillator consists of an inductor L and a capacitor C, building a parallel resonance tank, and an active element -R, compensating the losses of the inductor [24]. The circuit results into an oscillator with angular center frequency given by:

$$\omega_c = \frac{1}{\sqrt{LC}} \qquad 2.25$$

Since the capacitance C is proportional to the tuning voltage input V_{tune}, and ω_c is proportional to the capacitance, ω_c is dependent on V_{tune} and this makes the oscillator a voltage controlled one.

The capacitor C in Figure 2.23 not only consists of a variable capacitor to tune the oscillator, but it also includes the parasitics or fixed capacitances of the inductor, the active elements, and of any load connected to the VCO (output driver, mixer, prescaler, etc.).

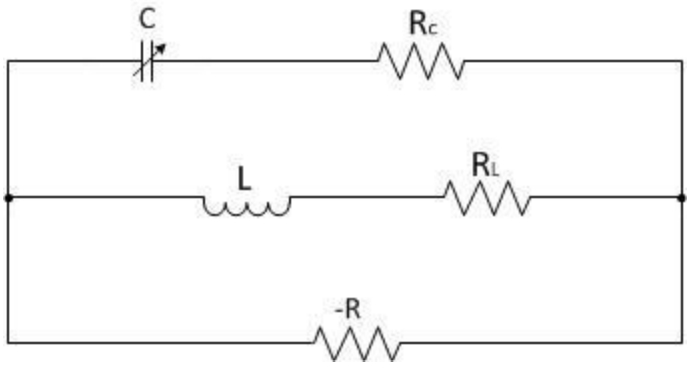


Figure 2.23 - Basic LC-VCO

In comparison to ring oscillators, LC oscillators have a rather limited tuning range, but feature lower phase noise at a lower power consumption. The area of an LC oscillator with an integrated coil is much bigger than the area of a ring oscillator.

Figure 2.25 shows examples of some LC oscillator topologies [28].

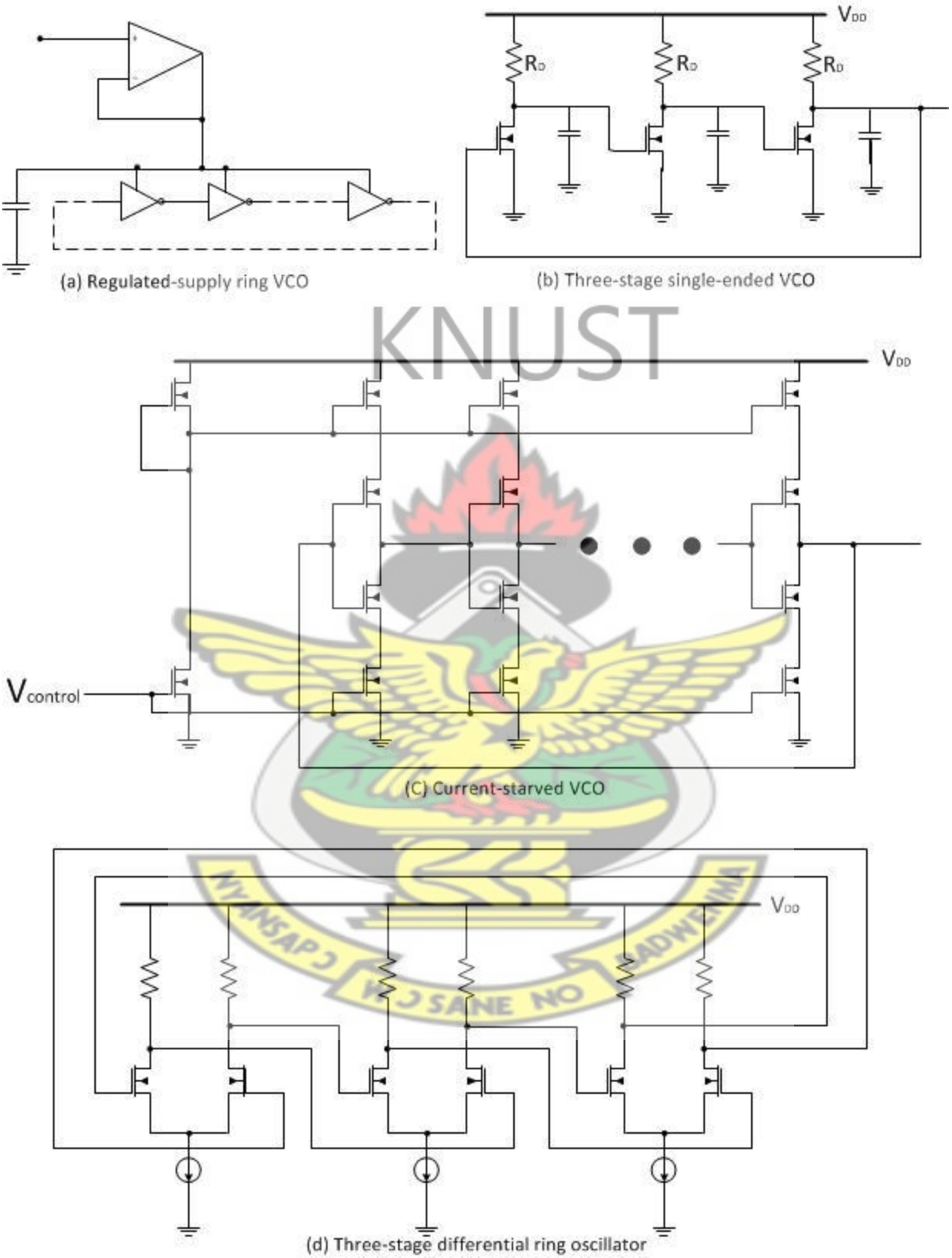


Figure 2.24 - Some ring VCO architectures

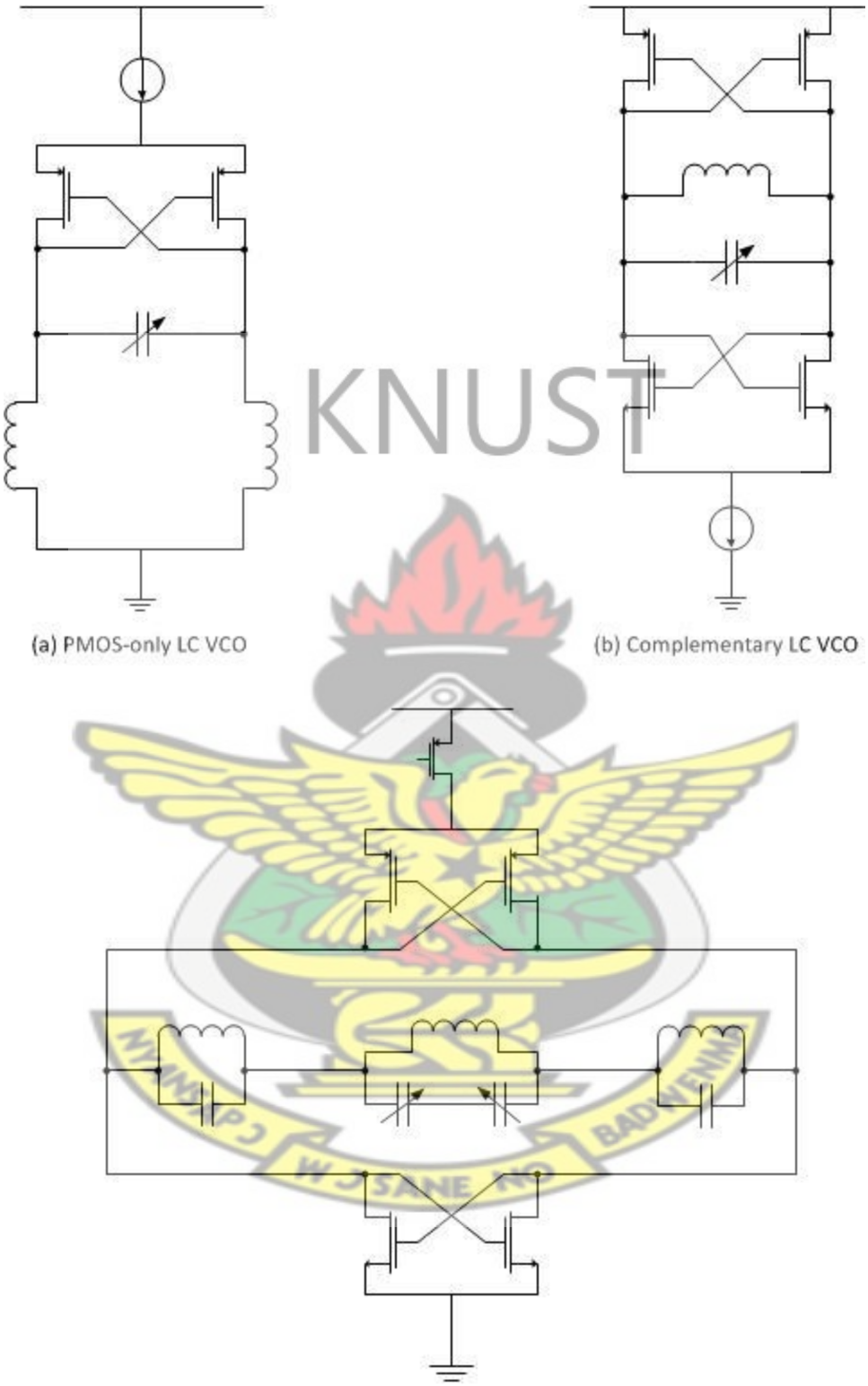


Figure 2.25 - Some LC VCO architectures

2.6.5 Loop Divider

As discussed in section 2.5, a loop divider is placed in the PLL feedback path to enable synthesis of a particular frequency from the reference frequency. Figure 2.26 demonstrates the basic operation of a loop divider using a loop division ratio of 4.

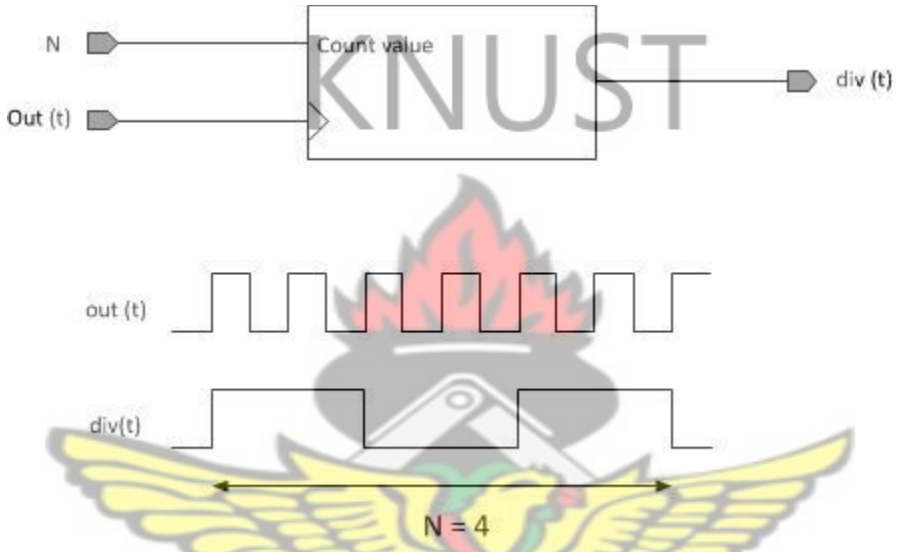


Figure 2.26 - Loop divider basic operation

The frequency of the feedback signal, ω_{fb} , is related to the output signal, ω_{out}, by the formula:

$$\omega_{fb}(t) = \frac{1}{N}\omega_{out}(t) \qquad \textbf{2.26}$$

In the most basic implementation, the frequency dividers are made up of logic gates and flip-flops. They can be grouped into synchronous and asynchronous types, depending on how the synchronization of these flip-flops is performed. In the synchronous dividers, each flip-flop is triggered by the input signal of the divider (clock). On the other hand, in the asynchronous dividers the input signal of the divider feeds the first flip-flop, which triggers the second and so on. Therefore, the synchronous frequency dividers achieve a complete transition faster than the asynchronous ones [2].

Perhaps the simplest frequency divider is the basic D flip-flop. If the $\bar{Q}$ output on a D-type flip-flop is connected directly to the D input giving the device closed loop "feedback", successive clock pulses will make the bistable "toggle" once every two clock cycles, essentially making it a divide by two loop divider. Figure 2.27 illustrates this concept.

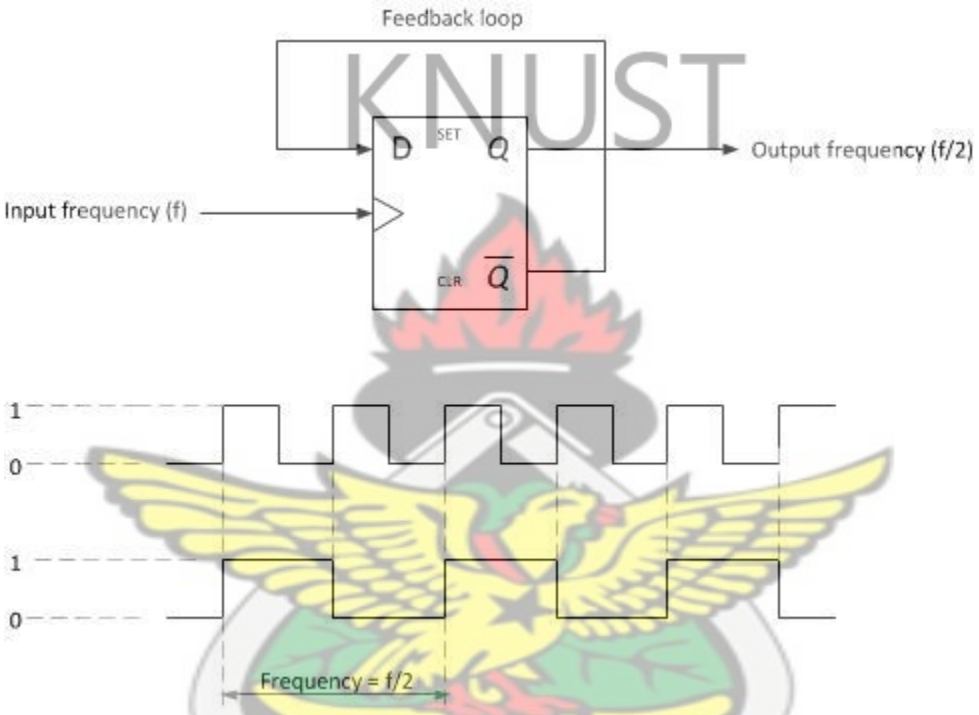


Figure 2.27 - The basic D flip-flop as a frequency divider

It can be seen from the frequency waveforms above, that by "feeding back" the output from Q to the input terminal D, the output pulses at Q have a frequency that are exactly one half (f/2) that of the input clock frequency, (F_{in}). In other words the circuit produces *frequency division* as it now divides the input frequency by a factor of two since Q = 1 once every two clock cycles.

The above D flip-flop can be arranged in “series”, either in an asynchronous or synchronous configuration to produce frequency dividers with higher division ratios of 4, 8, 16, 32, and so on.

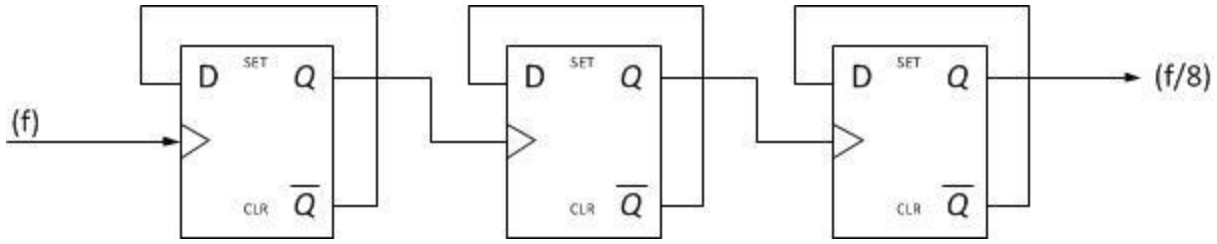


Figure 2.28 - Asynchronous divide by 8 using D flip-flops

The primary advantage of asynchronous dividers is that since each stage runs at a reduced frequency, there is lower power consumption. This also means that there is less load on the high frequency clock. However asynchronous dividers have higher jitter accumulation [13].

Figure 2.29 shows a synchronous frequency divider architecture [29]. Synchronous dividers accumulate less jitter, which is a desirable characteristic. However because each flip-flop stage runs at the maximum frequency, they tend to consume more power. There is also a large load on the high frequency clock [13].

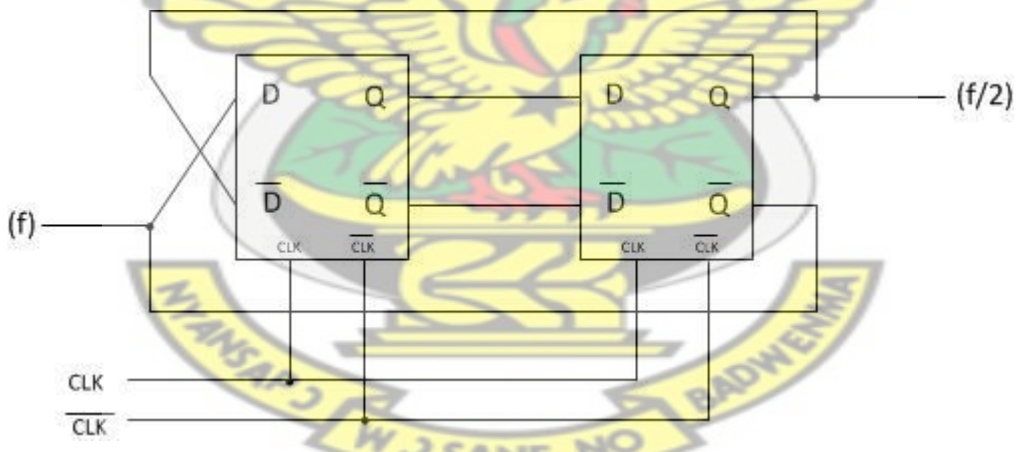


Figure 2.29 - Synchronous divide by 2

2.7 Built-In Self-Test

According to the SEMATECH Official Dictionary, BIST is *"any of the methods of testing an integrated circuit (IC) that uses special circuits designed into the IC. This circuitry performs test functions on the IC, and signals whether the parts of the IC covered by the BIST circuits are working properly."*

Hence, the basic concept of BIST involves the design of test circuitry around a system that automatically tests the system by applying certain test stimulus and observing the corresponding system response. Because the test framework is embedded directly into the system hardware, the testing process has the potential of being faster and more economical than using an external test setup.

2.7.1 The Need for BIST

To understand the need for BIST one needs to be aware of the various testing procedures involved during the design and manufacture of any system. There are three main phases in the design cycle of a product where testing plays a crucial role:

i. Design Verification: In this phase the design is simulated with respect to logic, switching levels and timing. This checks if the design satisfies the system level specification.

ii. Testing for defects introduced by the manufacturing process: This consists of wafer level testing and device level testing. In the former, a chip on a wafer is tested and if passed, is packaged to form a device which then gives rise to device level testing.

iii. System Operation testing: A system may be implemented using a chip-set where each chip takes on a specific system function. During fabrication at the board level however, printed circuit board (PCB) faults may be introduced. The system needs to be checked for such faults. Concurrent fault detection circuits (CFDCs) and error correction codes such as parity of cyclic redundancy check (CRC), are used to determine the existence of such faults [12].

The number of transistors contained in most VLSI devices has been increasing exponentially. In addition there has been an ever increasing functional complexity in ICs in general [7]. All these developments make the individual chips on a system less accessible from the point of view of testing, making testing a big challenge. With increasing device sizes and decreasing component sizes, the number and types of defects that can occur during manufacturing increase drastically, thereby increasing the cost of testing [30].

The ability to provide some level of fault diagnosis (information regarding the location and possibly the type of the fault or defect) during manufacturing testing is needed, and that is where BIST comes in. BIST can partition the device into levels and then perform testing. It therefore offers a hierarchical solution to the testing problem such that the burden on the system level test is reduced.

Figure 2.30 presents a block diagram of the basic BIST hierarchy. The test controller at the system level can simultaneously activate self-test on all boards. In turn, the test controller on each board activates self-test on each chip on that board. The pattern generator produces a sequence of test vectors for the circuit under test (CUT), while the response analyzer compares the output response of the CUT with its fault-free response [31].

2.7.2 Basic BIST Architecture

The basic BIST architecture requires the addition of three hardware blocks to a digital circuit: a pattern generator, a response analyzer, and a test controller. This architecture is illustrated in Figure 1.1.

Test Pattern Generator

Depending upon the desired fault coverage and the specific faults to be tested for, a sequence of test vectors (test vector suite) is developed for the CUT. It is the function of the TPG to generate these test vectors and apply them to the CUT in the correct sequence. Examples of pattern generators are a ROM with stored patterns, a counter, and a linear feedback shift

register (LFSR).

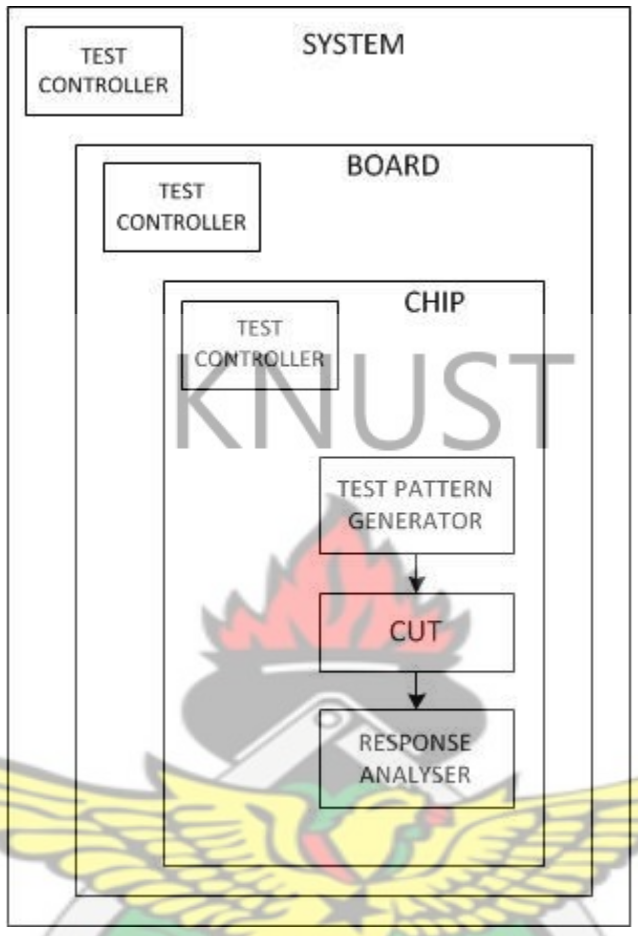


Figure 2.30 - Basic BIST hierarchy

BIST/Test Controller

The BIST controller controls all the transactions necessary to perform self-test. In large or distributed BIST systems, it may also communicate with other test controllers to verify the integrity of the system as a whole [32]. The importance of the controller is clearly seen from its many outgoing connections shown in Figure 1.1. The external interface of the test controller consists of a single input and single output signal. The test controller's single input signal is used to initiate the self-test sequence. The test controller then places the CUT in test mode by activating input isolation circuitry that allows the test pattern generator (TPG) and controller to drive the circuit's inputs directly. The isolation circuitry could be a multiplexer or some other switching system. Depending on the implementation, the test controller may also be

responsible for feeding the TPG with inputs which are used to generate the correct output sequence. During the test sequence, the controller interacts with the output response analyzer to ensure that the proper signals are being compared. To accomplish this task, the controller may need to know the number of shift commands necessary for scan-based testing [33].

Output Response Analyzer

The response of the system to the applied test vectors needs to be analyzed and a decision made about the system being faulty or fault-free. This function of comparing the output response of the CUT with its fault-free response is performed by the ORA. The ORA compacts the output response patterns from the CUT into a single pass/fail indication. Response analyzers may be implemented in hardware by making used of a comparator along with a ROM based lookup table that stores the fault-free response of the CUT [33].

2.7.3 Test Pattern Generation

The generated test patterns applied to the CUT directly determine the extent of fault coverage obtained. Some basic TPG implementation techniques used in BIST approaches are now discussed.

Exhaustive and Pseudo-exhaustive Test Patterns

Exhaustive pattern BIST eliminates the test generation process and has very high fault coverage [33]. To test an n-input block of combinational logic, all possible 2^n input patterns are applied to the block. Thus, the detection of all combinational faults is guaranteed. However, faults that may cause combinational circuits to exhibit sequential behavior (for instance, CMOS stuck-open faults) may not be detected. Also even with high clock speeds, the time required to apply the patterns may make exhaustive-pattern BIST impractical for circuits with n greater than about 25 [30].

More practically, circuits can be segmented into smaller possibly overlapping units. Each block

is then exhaustively tested. This approach is referred to as pseudo-exhaustive testing.

Deterministic Test Patterns

These test patterns are developed to detect specific faults and/or structural defects for a given CUT. The deterministic test vectors are stored in a ROM and the test vector sequence applied to the CUT is controlled by memory access control circuitry. This approach is often referred to as the "stored test patterns" approach [33].

Algorithmic Test Patterns:

Like deterministic test patterns, algorithmic test patterns are specific to a given CUT and are developed to test for specific fault models. Because of the repetition and/or sequence associated with algorithmic test patterns, they are implemented in hardware using finite state machines (FSMs) rather than being stored in a ROM like deterministic test patterns [33].

Random Test Patterns

In large designs such as a microprocessor, the number of possible states is so large that it is not feasible to generate all possible input vector sequences. A truly random test vector sequence is used for the functional verification of these large designs. However, the generation of truly random test vectors for a BIST application is not very useful since the fault coverage would be different every time the test is performed as the generated test vector sequence would be different and unique every time [32].

Pseudo-Random Test Patterns

These are the most frequently used test patterns in BIST applications [34]. Pseudo-random test patterns have properties similar to random test patterns, but in this case the vector sequences are repeatable. The repeatability of a test vector sequence ensures that the same set of faults is being tested every time a test run is performed. In general, pseudo random testing requires more patterns than deterministic patterns, but much fewer than exhaustive testing [32]. Linear Feedback Shift Registers (LFSRs) are the most commonly used hardware implementation methods for pseudo-random TPGs [35].

2.7.4 Response Analysis

Hardware structures that may be employed to determine the validity of a CUT's outputs include ROM and comparison logic, LFSRs, Multiple-input signature register (MISR), Cellular automaton, level counters, transition counters, and XOR trees.

Generally, response analysis refers to comparing the BIST output with the known output to determine whether the circuit is faulty or fault-free. One way to do this is to store the correct sequence on a ROM and use comparison logic. Alternatively, the response can be compacted into a relatively short binary sequence referred to as a signature.

After obtaining a response sequence R for a given order of test vectors from a gold CUT or a simulator, a compaction function *C* is used to produce a vector or a set of vectors *C(R).* Due to compaction, the number of bits in *C(R)* will be fewer than the number in *R.* The compacted vectors are stored on chip or off chip. During BIST, the compaction function *C* is used to compact the CUT's actual responses *R* to provide $C(R^*)$. Finally, to determine the CUT's status *C(R)* and $C(R^*)$ are compared. The CUT is declared faultless if these match.

For compaction to be of practical value, the function *C* should be simple to implement on chip, the compacted responses should be sufficiently small, and, above all, a faulty CUT should not be declared fault-free. If a faulty circuit and the fault-free circuit provide different response sequences but the compacted response sequences are identical *aliasing* is said to have occurred [31].

Three common compaction functions are the transition count, the signature analysis and the syndrome or 1's counting functions. All three are prone to aliasing. The most popular compaction function is signature analysis which is implemented using either an LFSR or MISR.

2.7.5 Advantages and Disadvantages of BIST

Advantages

i. Vertical Testability: The same testing approach could be used to cover wafer and device level testing, manufacturing testing as well as system level testing in the field where the system operates.

ii. Reduction in Testing Costs: Restated, the inclusion of BIST in a system design minimizes the amount of external hardware required for carrying out testing significantly.

iii. In-Field Testing capability: Once the design is functional and operating in the field, it is possible to remotely test the design for functional integrity using BIST, without requiring direct test access.

Disadvantages

i. Area Overhead: The inclusion of BIST in a particular system design results in greater consumption of die area when compared to the original system design. This may seriously impact the cost of the chip as the yield per wafer reduces with the inclusion of BIST.

ii. Performance penalties: The inclusion of BIST circuitry adds to the combinational delay between registers in the design. Hence, with the inclusion of BIST the maximum clock frequency at which the original design could operate will reduce, resulting in reduced performance.

iii. Additional Design time and Effort: During the design cycle of the product, resources in the form of additional time and man power will be devoted for the implementation of BIST in the designed system.

iv. Added Risk: What if the fault existed in the BIST circuitry while the CUT operated correctly. Under this scenario, the whole chip would be regarded as faulty, even though it could perform its function correctly.

[34], [31], [33].

The advantages of BIST outweigh its disadvantages. As a result, BIST is implemented in a majority of the electronic systems today.

2.7.6 A PLL Test Strategy

The description of PLL functions that have so far been covered in this text suggests that PLL testing may be achieved by exercising the PLL under test (PUT) as described in [11].

In [11], the PLL is subjected to testing by making use of the following unique checks.

i. Checking the start sequence: This is the verification that the PLL can be brought up to frequency within a specified time from the time the system is started. This includes the verification of phase locking.

ii. Normal mode operation: After phase lock, checks must be done to ascertain the output signal of the PLL is at the desired frequency.

iii. Changing frequency dynamically: This is the verification of the fact that the PLL can operate in various modes such as system test mode, normal operation mode, etc. It may involve reprogramming of the PLL in order to effect frequency change dynamically.

iv. Checking the "stop" sequence: This is the verification that the PLL can be stopped when a specific sequence of signals is applied to the requisite inputs. This, together with the start sequence check, verifies that the driven chips would cease and resume operation in tandem.

This strategy is a defect-oriented method although it takes advantage of the function and dynamics of various components and the entire PLL circuitry to achieve testing. The test is first initialized and then the entire PLL is exercised to test all the components and their interactions. Responses are collected in the form of frequency variations.

According to experimental simulations carried out in [11] using *Spectre®,* the described BIST scheme is very effective in detecting faults. A MOS transistor level fault model was used for both digital and analog components during simulations. Drain open, source open, gate-to-drain short, gate-to-source short, and transistor stuck-on faults were modeled. In all of the 316 faults simulated, only 8 faults remained undetected. All the undetected faults were later found to be redundant.

Chapter 3

A 40-100MHz Phase-locked Loop Frequency Synthesizer with Built-In Self-Test

3.0 Introduction

This section describes the development of a phase-locked loop based frequency synthesizer with integrated built-in self-test. The chapter is organized into two main parts. The first section describes the design of the frequency synthesizer at the transistor level. All implemented circuits along with their layouts are presented. Factors informing choices and compromises made are explained. The second part details the design of the PLL BIST circuit.

3.1 Fully Integrated PLL Design Procedure

The procedure used for the design of the phase-locked loop is presented here. This procedure is adopted from [17] and modified slightly. The procedure basically describes how to define system level parameters such as the value of the filter capacitances and VCO gain for a fully integrated charge pump PLL. It is described he using a prototype PLL with variables for the system level parameters.

i. Specify VCO Tuning Range

The VCO output tuning range essentially becomes the PLL's frequency range. It refers to the range of frequencies under which the VCO (and hence the PLL) is operating. The frequency range for the sample PLL is $F_{min}\ to\ F_{max}$. This requires that the VCO have a tuning range at least $F_{min}\ to\ F_{max}$. However to achieve the frequency extremes of F_{min} and F_{max}, the

actual range must be made slightly wider.

$$VCO\ maximum\ frequency\ \geq\ F_{max} \tag{3.1}$$

$$VCO\ minimum\ frequency\ \leq\ F_{min} \tag{3.2}$$

ii. Specify the loop division ratio

The loop division ratio, N, for the prototype PLL is chosen to be 32. A division ratio of 32 is a typical value used in frequency synthesizers.

$$N = 32$$

iii. Choose the damping factor, ζ_{mean}

With reference to Figure 2.6, it is easily seen that choosing a damping factor between 0.5 and 2 will generally produce an optimal response. A value of 0.7 is commonly used, and serves as good compromise between speed and stability.

$$\zeta_{mean} = \frac{1}{\sqrt{2}}$$

iv. Determine the natural frequency, ω_n

The natural frequency has a significant effect on the loop bandwidth. For a charge pump PLL with a passive loop filter, the loop bandwidth, ω_{3dB} , is related to the natural frequency by equation (2.17. Given $\zeta_{mean} \approx 0.707$ from step three above,

$$\omega_n = \frac{\omega_{3dB}}{\left[2\zeta^2 + 1 + \sqrt{(2\zeta^2 + 1)^2 + 1}\right]^{1/2}}$$

Hence the natural frequency is given by:

$$\omega_n = \frac{\omega_{3dB}}{2.06} \quad \textbf{(3.3)}$$

Ideally, the loop bandwidth should be less than 1/10 of the input reference frequency (ω_{ref}) in order to avoid the continuous time approximations of the charge pump PLL breaking down. However, it is also preferred that the bandwidth be as wide as possible, as this suppresses VCO phase noise, which is the dominant source of noise in integrated PLLs. As a tradeoff between noise and stability the loop bandwidth is made 75% less than 1/10 of the input reference frequency. The loop bandwidth is then given by:

$$\omega_{3dB} = (0.75)\frac{\omega_{ref}}{10} = 0.075\omega_{ref}\, {}^{rad}/_{s} \quad \textbf{(3.4)}$$

From equation (3.3, the natural frequency for $\zeta = 0.707$ then becomes the following.

$$\omega_n = \frac{0.075\omega_{ref}}{2.06} rad/s \quad \textbf{(3.5)}$$

v. Determine the VCO gain

The gain of the VCO depends on the frequency tuning range and control voltage range of the VCO. From (i), the VCO tuning range was determined to be F_{min} to F_{max}. The VCO control voltage range is limited by the power supply and the voltage levels necessary to keep the charge pump in saturation. The charge pump will no longer behave ideally if the transistors which form the current source move out of saturation as a result of the VCO control voltage falling too low or rising too high. Assuming a control voltage range of V_{tuning}, the VCO gain may be calculated as:

$$K_{VCO} = \frac{2\pi(F_{max} - F_{min})}{V_{tuning}} {}^{rad}/_{sV} \quad \textbf{(3.6)}$$

vi. Set charge pump current and determine loop filter primacy capacitor value

The value of the primary loop filter capacitor can be calculated from the following formula derived from equation (2.16.

$$C_1 = \frac{IK_{VCO}}{2\pi N{\omega_n}^2} \quad \textbf{(3.7)}$$

Generally, a higher charge pump current is preferred because a higher charge pump current equals a higher loop gain which results in a more stable system. The downside, as can be seen from equation (3.10 above is the direct proportionality between the loop filter capacitor C_1 and the charge pump current I. Increasing the charge pump current will produce a larger capacitor which will occupy a larger die area. The charge pump current is chosen such that it produces a decent loop gain, without producing a capacitor that is too large. This is easily done through a short iterative calculation process.

vii. Determine secondary capacitor and resistor values

In section 2.6.3, the need for a resistor to improve stability as well as a secondary capacitor to suppress glitches in the loop filter was explained. The value of this resistor R, is determined by the following relation.

$$R = \frac{2\zeta}{\omega_n C_1} \quad \textbf{(3.8)}$$

The general convention is to make the secondary capacitor C_2 less than a tenth of the primary capacitor C_1. This way C_2 may be neglected in the mathematical analysis of the loop so that it may still be considered as a second order system. Hence C_2 is given by:

$$C_2 < \frac{C_1}{10} \quad \textbf{(3.9)}$$

This seven step design procedure has defined all the system level parameters that are needed to start the design. Often during the design process though, it may be possible to optimize some of these values through simulations. The transistor level design can now be started.

3.2 PLL Transistor Level Design

All implemented circuits along with their layouts are presented here. Circuits are first designed and tested using CADENCE IC. When simulation results meet required specifications, corresponding layouts are developed. The layout is accomplished with good matching and high-frequency techniques. All simulations and layouts are done using the same tool. The integrated circuit is prepared for fabrication through the MOSIS foundry using the AMI 0.6µ CMOS process.

The final circuit design is a mixed signal one. Some blocks, such as the PFD are obviously digital circuits, while other blocks like the VCO are purely analog. Generally, the digital blocks are much easier to describe and design. The analog blocks are a bit trickier to design. Relevant calculations are done to obtain circuit parameters and values. However, as is most often the case in analog design, these will not necessarily represent the optimum values that may be used. Hence, it is often necessary to design based on calculated values, and subsequently optimize whatever values are obtained through simulations. This approach is used. Formulas and calculations made will be shown and where relevant, test circuitry used to parameterize will be presented as well. Block designs will be presented in order of relevance to other designs. Hence if parameters from the VCO are needed to design the charge pump, then the VCO design will be described before that of the charge pump.

The threshold voltage and transconductance technology parameters for this particular process are first determined using a simple characterization procedure. The most important results are repeated in the table below for emphasis. These values will be quoted and used throughout the design process without repeated reference. Transistor W/L ratios as well as specific sizes will be given. Unless otherwise stated, the channel should be taken to be the minimum process length of 0.6µm.

Table 3.1 - AMI 0.6u process technology parameters

Transistor Type	V_T (threshold voltage)	$K = {\mu_n c_{ox}}/{2}\ ({\mu A}/{V^2})$
PMOS	0.92	19.2
NMOS	0.67	60

3.2.1 Voltage Controlled Oscillator

From section 2.6.4 it was established that a simple chain of an odd number of inverters may oscillate, but that oscillation frequency is fixed. By employing a technique that makes it possible for the oscillation frequency to be changed via a voltage, a voltage controlled oscillator can be designed. The simplest way to control the charge and discharge time of an inverter is to control the current through the inverter, via a voltage controlled current source as shown in Figure 3.1 (See also Figure 2.24(c)). This current source is driven by the control voltage, V_{cntl}, and will determine the charge up and discharge time of the inverter. This topology is called a current-starved inverter, as the regular inverter is short of the current it would normally consume. It forms the core of the designed VCO.

With correct sizing and current levels, an odd number of stages of these current starved inverters can make a good VCO. Seven of these stages are employed in designing a VCO with a tuning range of about 40 – 100MHz. To achieve the frequency extremes, the actual range is made wider, about 30 – 105MHz. The complete VCO schematic is shown in Figure 3.3. This design is simple and the oscillation frequency can be achieved in a reasonably short amount of time. It also provides a wide tuning range due to the square law change in current levels in the footer device [36]. The tuning frequency range is increased by increasing the sizes of transistors. The final transistor sizes are determined through simulations in *Spectre®*. Table 3.2 shows the final width per length ratios.

The output waveform of the VCO can be improved by loading the output of the inverter with a capacitor. This however decreases oscillation frequency as it increases the charge up and discharge time of the inverting stage. A trade-off between the oscillation frequency and output waveform is necessary. This is done through simulation. The final capacitance value is chosen to be 200fF.

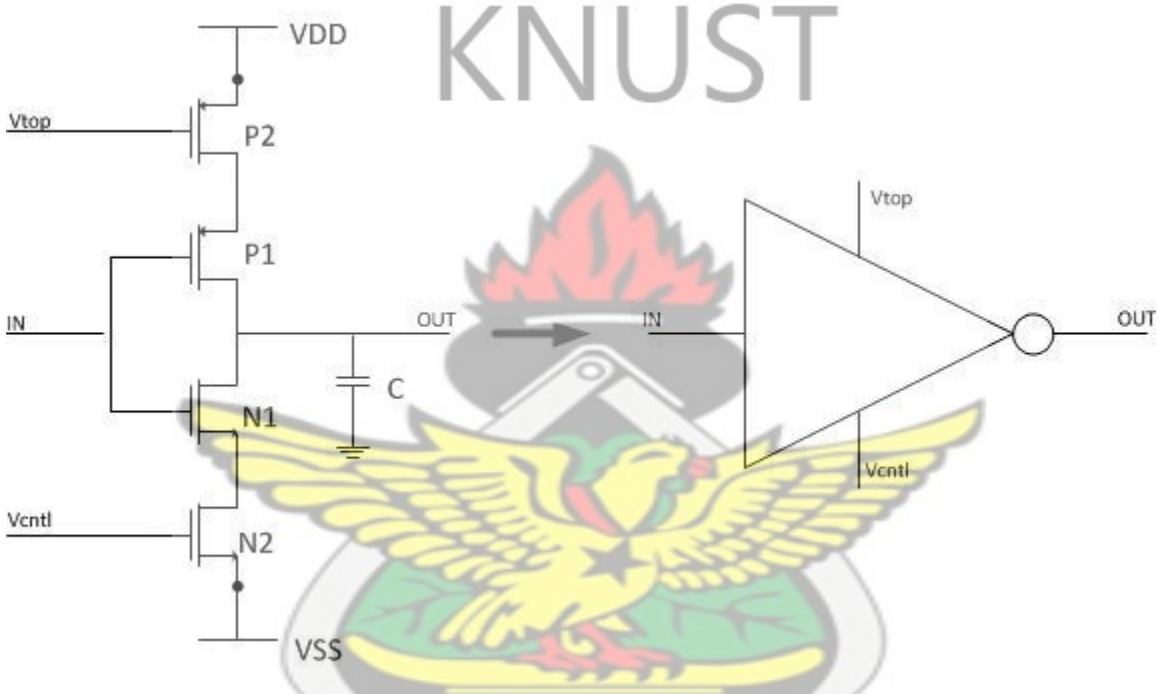


Figure 3.1 - VCO inverting stage

Table 3.2 - VCO inverting stage transistor sizes

Transistor	Size (W/L)
P1	62.5
P2	100
N1	37.5
N2	30

Figure 3.2 displays a graph showing the simulation results of an output frequency sweep of the VCO. From this graph, the K_{VCO} may be obtained. K_{VCO} is found by calculating the slope of the graph within the 'effective' region. In this context the 'effective' region simply refers to the area within the chosen F_{max} and F_{min}, shown in the graph as M1 and M0 respectively. Since this 'effective' region is non-linear, a good approximation of the gradient is found by breaking the region into shorter linear sections, calculating the slope of these, and finding an average for all calculated gradients. A more accurate K_{VCO} is found by using as many of these smaller strips as possible. From the graph shown in Figure 3.2 K_{VCO} is calculated to be 290Mrad/s.

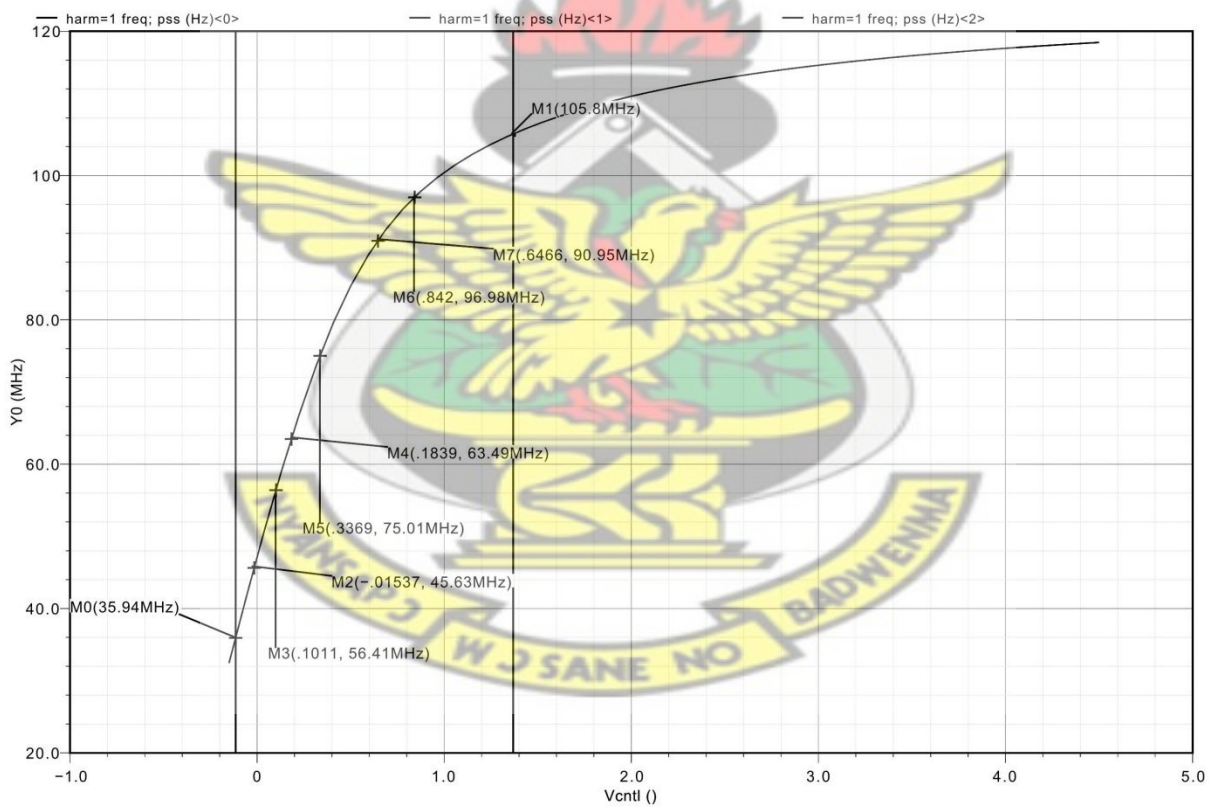


Figure 3.2 - VCO output frequency plot

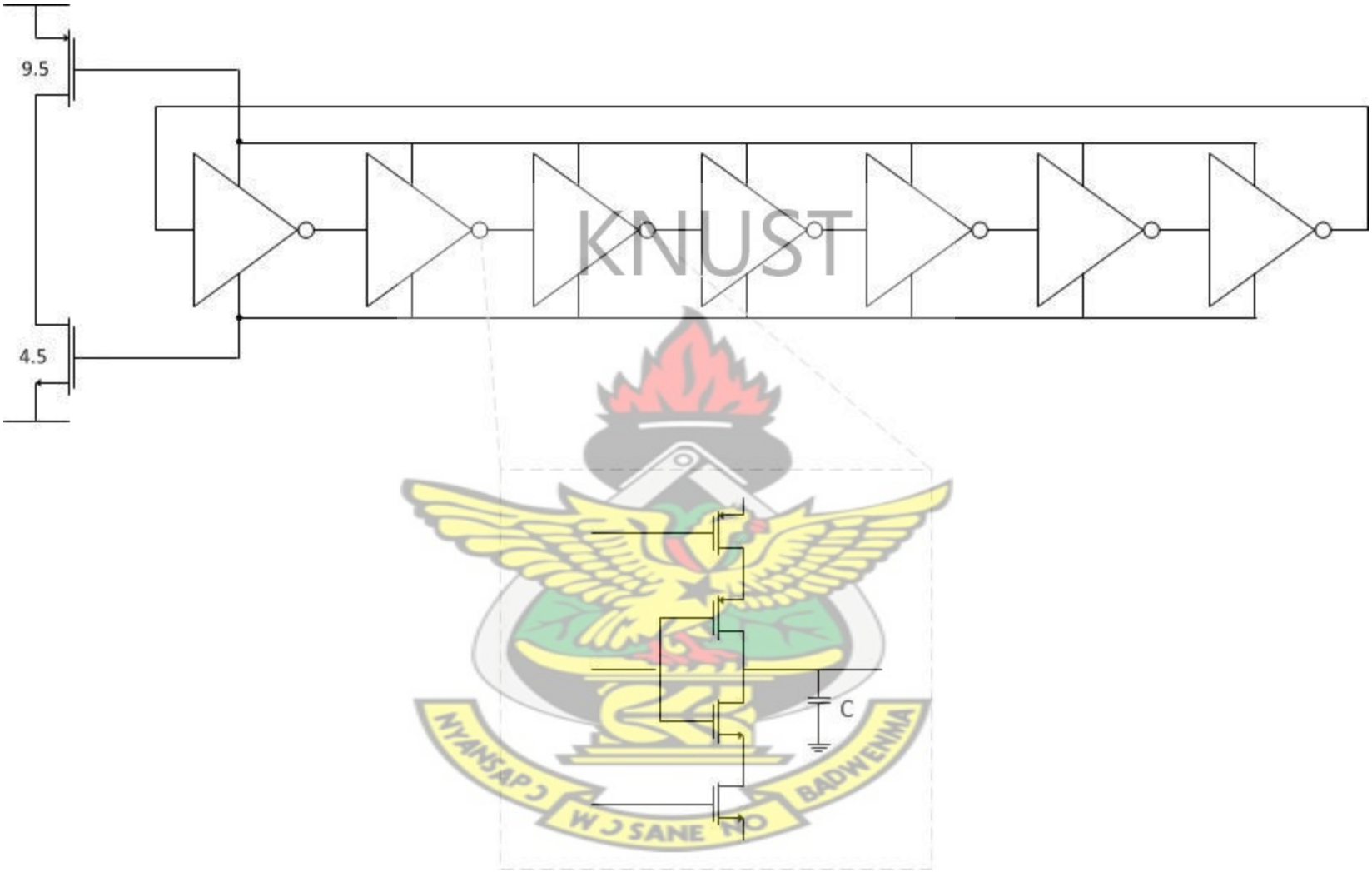


Figure 3.3 - Complete VCO schematic

3.2.2 Phase/Frequency Detector

A simple but fully functional phase/frequency detector was described in section 2.6.1. Although this circuit proves quite useful in some practical applications, the more generally used design is the circuit shown below [37].

This conventional sequential frequency and phase detecting logic is chosen because of its lower sensitivity to duty cycle [38]. This circuit can also operate at very high frequencies since the critical path is limited by just three gate delays: two from the cross-coupled two input NANDs and one from the four input RESET NAND. For these reasons, in this process, the PFD is implemented as shown in Figure 3.4.

The traditional design of NAND gates in CMOS, made up of series-parallel combinations of MOSFETS is used in implementing the two-input, three-input and four-input NAND gates. The circuit diagrams for these are shown in the appendix.

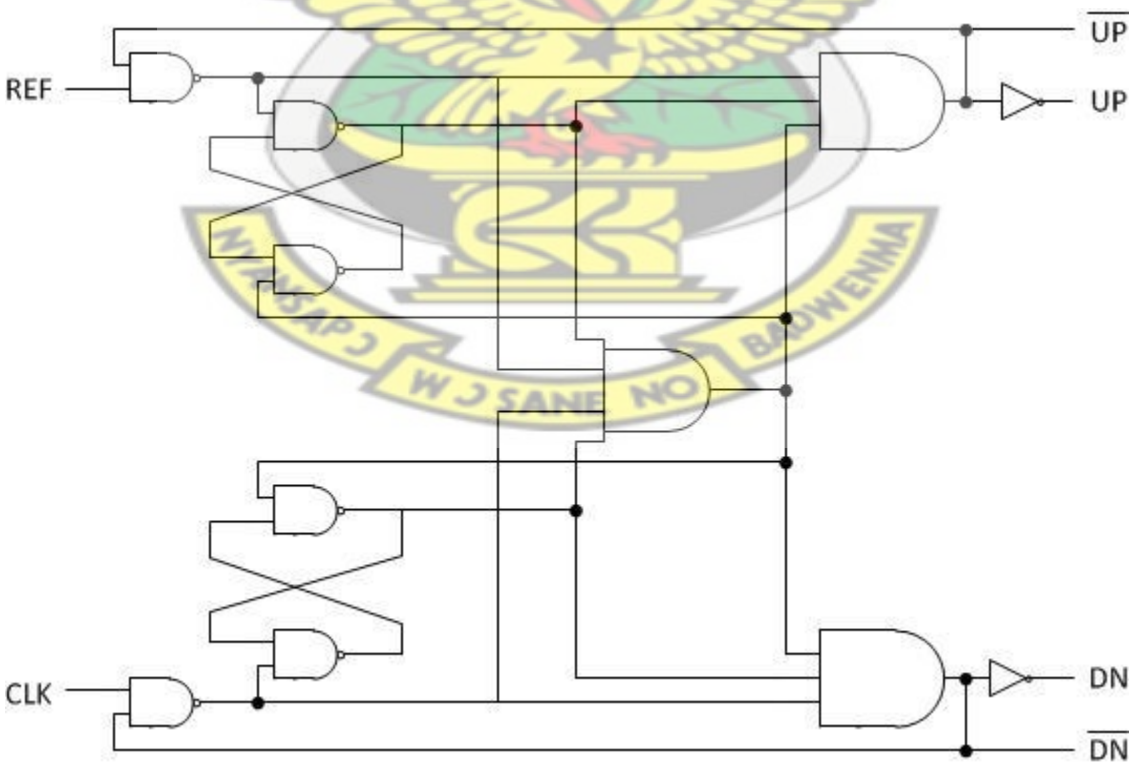


Figure 3.4 - The Phase/Frequency detector

3.2.3 Charge Pump

The charge pump designed in this system is shown in Figure 3.5 [39]. It is only a slight variation of the basic single-ended charge pump first shown in Figure 2.17. Current mirrors are used to implement the current sources and the switches are implemented with transistors which turn on or off depending on the gate voltage. The matching switches could lead to the confusion of this design with differential charge pump designs. However, this design is rather evidently a single-ended one as it has only one output and therefore requires only one loop filter.

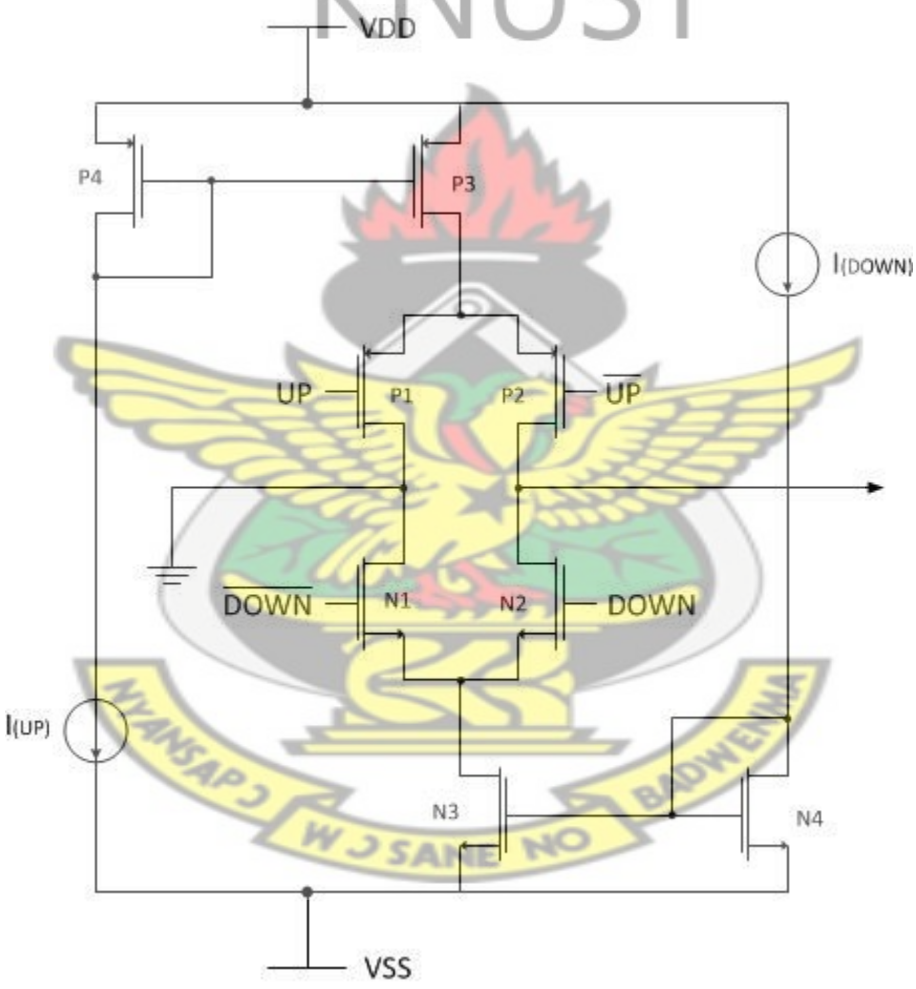


Figure 3.5 - Designed charge pump

The PMOS current mirror is one to one current mirror consisting of the diode connected transistor P4 as well as a dimensionally matching transistor P3. It is used to mirror I_{UP} into the charge pump. This I_{UP} current either goes into the loop filter or into the ground node depending

on the position of the two PMOS switches, P1 and P2. The NMOS current mirror is one to one as well, and comprises N3 and N4 to mirror I_{DOWN} into the charge pump. The current I_{DOWN} either discharges the loop filter or pulls current from the ground node depending on the position of the two NMOS switches N1 and N2.

To ensure that the phase detector gain is constant during either charging or discharging of the loop filter, the same value of current must be used for both I_{UP} and I_{DOWN}. This value is chosen to be 25µA.

Determining transistor sizes

The VCO will have a tuning range of $1.3V$. With a supply of -$1.5V$ to +$1.5V$, (i.e. $3V$), the compliance voltage to be met is calculated as follows:

$$compliance\ voltage = (3 - 1.3)/2$$
$$= 0.85V$$
$$= 850mV\ per\ each\ rail$$

The appropriate V_{DSAT} needs to be chosen for the transistors P4 and N3 to satisfy this compliance range. The VCO has a tuning range of $1.3V$ and swings from a minimum voltage of $-0.4V$ to a maximum voltage of $+1.0V$. Hence, a V_{DSAT} of $0.5V$ is chosen to prevent the two transistors from moving out of saturation over the entire VCO tuning range. To verify this behavior, consider when the VCO input voltage drops to the minimum of -$0.4V$:

$$minimum\ drop\ across\ N3 = -0.4 - (-1.5)$$
$$= 1.1V \gg 0.5V$$

Hence, N3 remains in saturation.

Also consider when the VCO input voltage rises to the maximum of $1.0V$:

$$minimum\ drop\ across\ P3 = 1.5 - 1.0$$
$$= 0.5V$$

This is just enough to keep P3 in saturation.

A V_{DSAT} of $0.5V$ also allows for a drop of $0.35V$ (i.e. compliance voltage – saturation voltage) across the 'switch' transistors. With this chosen V_{DSAT} the minimum sizes for P3 and N3 are calculated as:

$$\frac{W}{L} = \frac{2I_D}{KV_{DSAT}{}^2} \qquad \textbf{(3.10)}$$

$\left(\frac{W}{L}\right)_{P3}$	$> \frac{2(25\mu)}{19.2\mu \times (0.5)^2}$ > 10.4

$\left(\frac{W}{L}\right)_{N3}$	$> \frac{2(25\mu)}{60\mu \times (0.5)^2}$ > 3.33

Since P3 and P4 form a one-to-one current mirror, P4, has the same dimensions as P3. Similarly, N4 has the same dimensions as N3.

The voltage drop across the switches should not exceed $0.35V$. To account for possible process variations however, a value of $0.2V$ will be used. The V_{gs} is then calculated as follows:

$$\begin{aligned} V_{gs} &= Supply - (drop\ across\ switches \\ &\qquad + drop\ across\ mirror) \\ &= 3 - (0.5 + 0.2) \\ &= 2.3V \end{aligned}$$

With this V_{gs} the minimum sizes for P1, P2 and N1, N2 are calculated as:

$$\left(\frac{W}{L}\right) = \frac{I}{K(|V_{gs}| - |V_t|)|V_{DS}|} \qquad \textbf{(3.10)}$$

$\left(\frac{W}{L}\right)_{P1,P2}$	$> \frac{25\mu A}{(19.2\,\mu A/V^2)(2.3-0.92)(0.2)}$ > 4.71

$\left(\frac{W}{L}\right)_{N1,N2}$	$> \frac{25\mu A}{(60\,\mu A/V^2)(2.3-0.67)(0.2)}$ > 1.28

The final transistor sizes after optimization in *Spectre®* simulations are given in Table 3.3.

Table 3.3 - Charge pump transistor sizes

Transistor(s)	Size (W/L)
P3,P4	12
N3,N4	5
P1,P2	5
N1,N2	2

The current sources in Figure 3.5 are designed using correctly sized resistors. The appropriate size of the resistors needed may be determined using Kirchhoff's current law.

To determine I_{UP} consider the loop in which it appears, on the left side of the charge pump. This loop is shown in Figure 3.6. By applying Kirchhoff's voltage law across the loop the value of R_{UP} can be determined. The voltage drop across P4 is the same as V_{ds}. V_{ds} is obtained through simulation. Both VDD and VSS are replaced with voltage sources of $1.5V$. A DC analysis is run to determine the DC operating point of P4, the results from which V_{ds} is easily obtained to be $1.266V$.

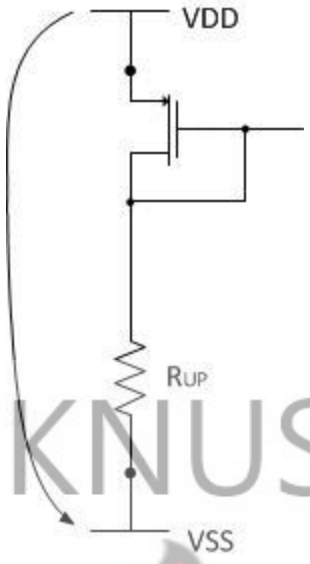


Figure 3.6 - Applying Kirchhoff's voltage law to the charge pump

Kirchhoff's voltage law is applied to the loop in Figure 3.6 as follows:

$V_{dd} - V_{ds} - IR_{UP} + V_{ss}$	$= 0$
$1.5 - 1.266 - 25\mu R_{UP} + 1.5$	$= 0$
R_{UP}	$= \frac{(3 - 1.266)V}{25\mu A}$
	$= 69.36K\Omega$

The same principle is applied to the loop containing VDD, $R_{DOWN,}$ N4, and VSS. The following equation is obtained and solved.

$V_{dd} - IR_{DOWN} - V_{ds} + V_{ss}$	$= 0$
$1.5 - 25\mu R_{DOWN} - 1.056 + 1.5$	$= 0$
R_{DOWN}	$= \frac{(3 - 1.056)V}{25\mu A}$
	$= 77.76K\Omega$

3.2.4 Loop Filter

The loop filter presented in chapter two is used. Section 2.6.3 explained the need for a resistor R, and a secondary capacitor C_2. The main requirement here is to determine the values of R, C_1 and C_2. The equations presented in section 3.1 are used. It is necessary however to first determine VCO gain, K_{VCO} and the natural frequency, ω_n. K_{VCO} was found in section 3.2.1 above from the output frequency plot of the VCO. A resolution of 3MHz is chosen for the frequency range of 35MHz to 105MHz. ω_n is then calculated from equation (3.5 as follows:

$$\omega_n = \frac{0.075\ (2\pi)(3)}{2.06}$$

$$= 0.6863\ {}^{M}/_{rad/_s}$$

If a value of 25μA is chosen for the charge pump current, then value of the main capacitor C_1 is also calculated from equation (3.7 as:

$$C_1 = \frac{25 \times 290}{2\pi \times 32 \times 0.6863^2}$$

$$= 76.6pF$$

The resistance R is determined from equation (3.8.

$$R = \frac{2 \times 0.707}{0.6863 \times 76.6}$$

$$= 26.90K\Omega$$

From equation (3.9, it is known that C_2 should simply be less than about 10% of C_1. A value of 7.5pf should suffice.

Figure 3.7 below shows the final loop filter design.

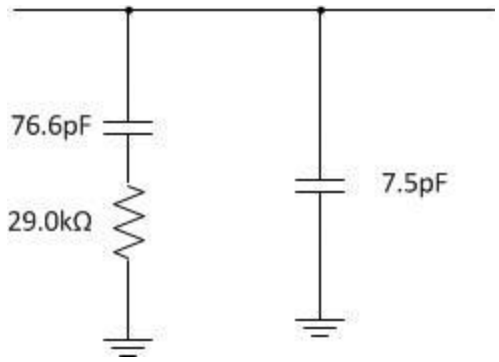


Figure 3.7 - Designed loop filter

3.2.5 Frequency Divider

The PLL frequency synthesizer is designed with a fixed loop division ratio of 32. A simple asynchronous binary counter, clocked by the input signal is used to create a power-of-2 integer divider. The loop divider is made with a series of flip-flops which have the $\bar{Q}$ output fed back to the D input. The initial signal to be 'divided' is fed to the clock of the first stage, while the Q outputs act as the clocks to subsequent stages. Each stage divides the frequency of the incoming signal by two, thus producing a 2^n for a divider with n stages, n being 5 in this case. Figure 3.8 shows this arrangement.

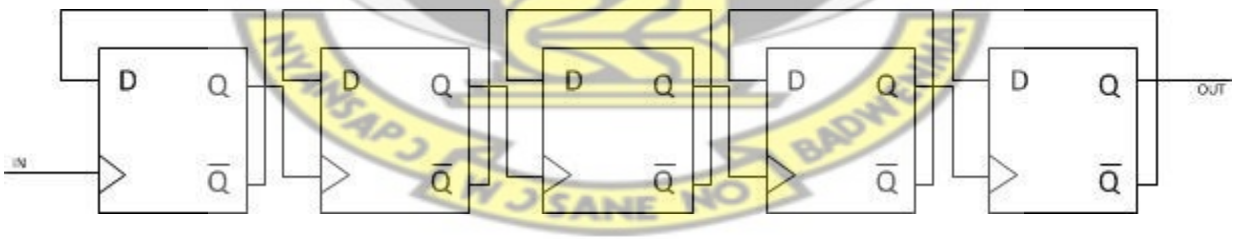


Figure 3.8 - Designed frequency divider

The flip-flops are implemented using true single phase clock latches. These are high frequency dynamic flip-flops designed for high speed and low power consumption. Figure 3.9 shows this implementation [40].

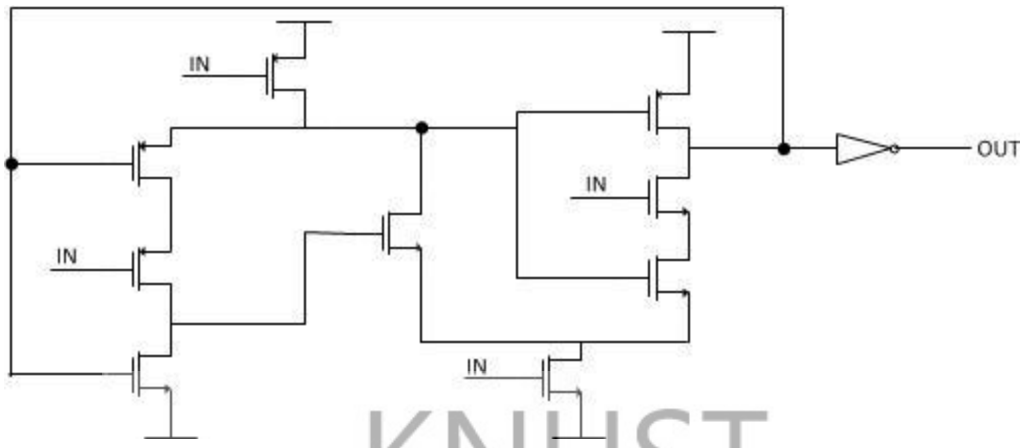


Figure 3.9 - D flip-flop for divider implementation

These flip-flops are capable of operating efficiently at frequencies of many gigahertzes. Since the highest frequency in this design is just 100MHz, it is possible to strip all the transistors down to their minimum sizes, while maintaining efficient operation within the desired frequency range. This is advantageous as it saves die area without compromising performance.

3.2.6 The PLL Frequency Synthesizer

Putting it all together, Figure 3.10 shows the final phase-locked loop frequency synthesizer design on a block level.

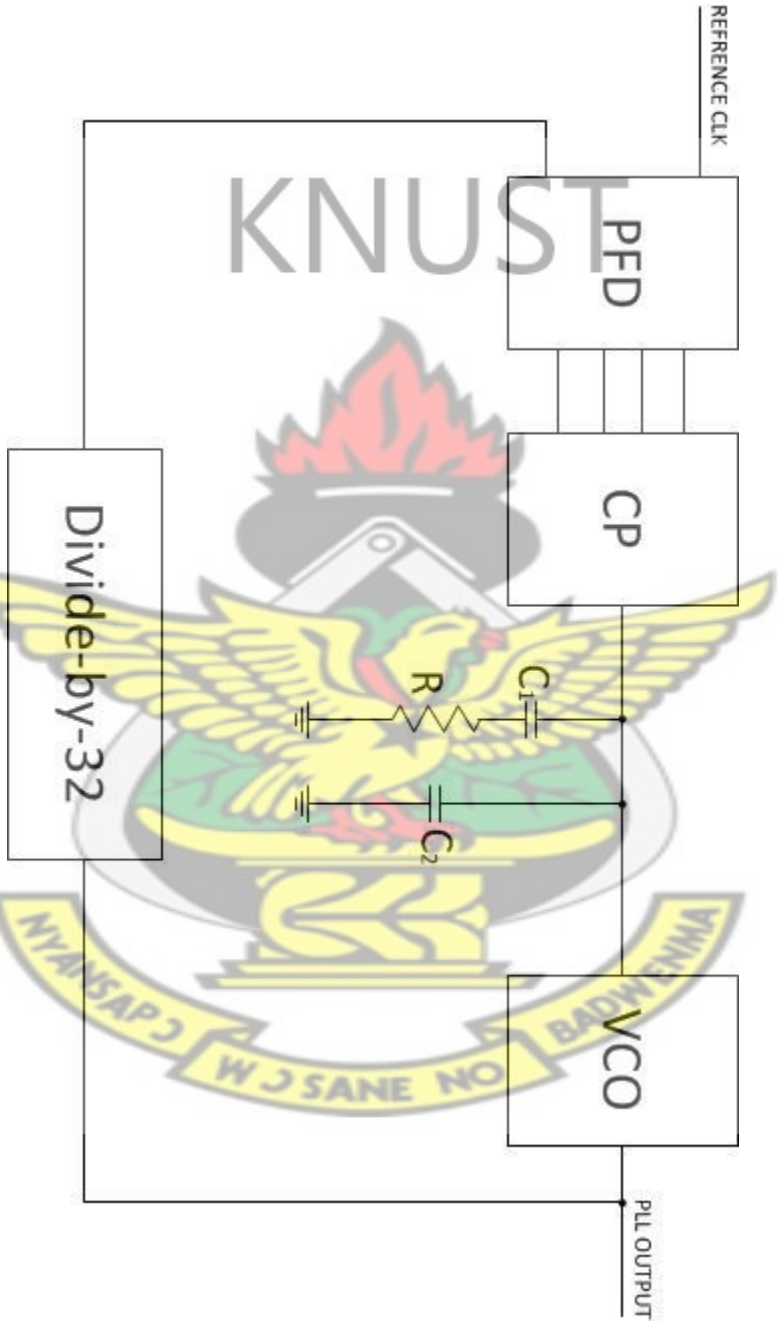


Figure 3.10 - Designed PLL frequency synthesizer

3.3 PLL BIST

A new charge-based frequency measurement BIST scheme is now presented. The BIST scheme presented here is an improvement over the scheme first presented in [11], to make use of less hardware.

In order to make the PLL testable a few modifications will have to be made to the design. The modifications come in the form of a little extra hardware, while keeping the original PLL hardware intact. Because the test strategy is implemented using minimal hardware and the output is purely digital, it is quite practical for the BIST scheme to be integrated on-chip.

3.3.1 Overview of Test Strategy

The BIST scheme implemented in this thesis is based on the same concepts described in [11]. The overall test concept is quite straightforward. The PLL is charged from a very low frequency to a known maximum frequency (F_{max}). The final frequency obtained is measured and compared to the known standard. With a little margin for variations, if the two values are the same the PLL is fault free. Otherwise it is faulty.

The test strategy is broken down into five stages below. Where applicable, the connection between a specific stage and the required checks presented in [11] is shown.

i. Initialization and "stop" sequence check: This involves the initial discharge of the PLL to a very low frequency to ensure that the next stage charges from a frequency as close as possible to the minimum VCO output frequency. This stage also discharges the charge pump and filter. The PLL is then pushed into a state where the output current is no longer AC but DC. This stage, if successful essentially stops the PLL operation and verifies the fourth "stop" sequence check described previously.

ii. Charge: In this stage, the PLL frequency is pushed from the minimum to a frequency as close as possible to the maximum output frequency by a steady increase in the

control voltage. This corresponds to the “start” sequence explained previously. The PLL frequency is held constant for a specified time afterwards, which also verifies phase locking.

iii. Measurement of F_{max}: The value of the obtained frequency after charge is measured. This is done by employing a digital counter to count the number of cycles within a specified time interval. This value is shifted out and compared with the expected value. Again, it should be pointed out that this frequency measurement is a test of the “normal mode operation” check presented in [11].

iv. Discharge: The output frequency is gradually decreased from F_{max} to a predefined frequency (F_{min}) using a steady decrease in the control voltage.

v. Measurement of F_{min}: The value of the obtained frequency after discharge is measured using the same method as in (iii) above. The obtained value is shifted out and compared with the expected value.

Figure 3.11 illustrates the test concept.

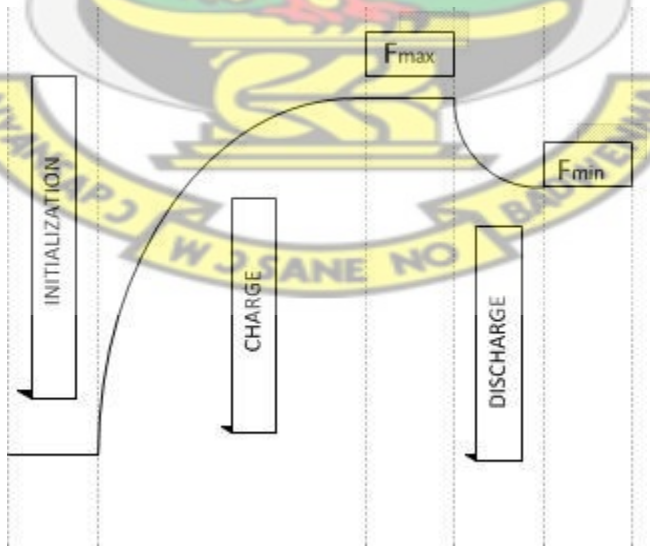


Figure 3.11 - Test Strategy Overview

3.3.2 Redesigning the PLL for testability

Figure 3.12 shows a system level PLL redesigned with BIST hardware to implement the concept illustrated in Figure 3.11 [11].

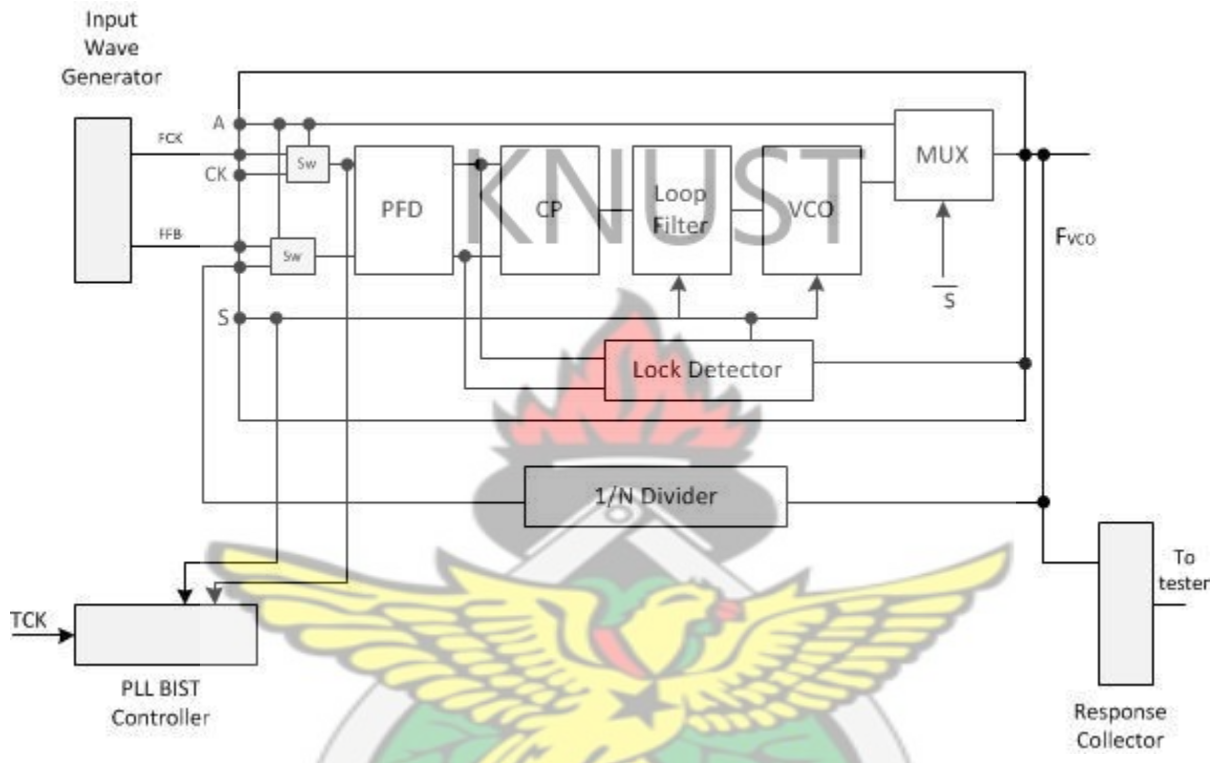


Figure 3.12 - PLL BIST

The BIST hardware consists of two switches Sw1 and Sw2 to select between the reference clock and test clock as well as the feedback signal and test feedback signal. A BIST Controller block with an input test clock is used to control the entire test process. The BIST Controller block is implemented as a counter whose outputs are selected in a particular combination to generate the test patterns and also to control the activation of the response collector. The Input Waveform Generator is a combinational circuit whose inputs are taken from the outputs of the BIST Controller. It generates the test patterns fed into the PLL to charge and discharge it. The response collector is a compactor implementing a predetermined compaction function. The 2-to-1 multiplexer (MUX) makes it possible to bypass the PLL with a clock signal at input A. S

serves as the control input for the multiplexer. The PLL can be bypassed by setting input S to 0 and clocking through input A. A zero-pulse on S resets the PLL to a hold state.

One issue with this BIST scheme is the amount of hardware. Even though the BIST hardware introduced is fairly minimal, it is possible to cut down on the used hardware even further.

The BIST controller is a counter. A counter is essentially a frequency divider. Since a frequency or loop divider is already part of the PLL circuitry, it should be possible to employ this loop divider as the BIST Controller.

This BIST scheme also requires that a reset mechanism be built into both the loop filter and the VCO. This means that PLLs that were not designed with a reset, which is quite often the case, have to be modified before they can be used with the BIST scheme. Besides, the reset mechanism introduces extra hardware and sometimes even complicates the VCO and filter design. By using the test pattern itself to initially discharge the PLL to a low frequency, the need for a reset can be eliminated. This also eliminates the need for the 2-to-1 multiplexer.

Taking these factors into account, Figure 3.13 shows the proposed BIST scheme. As seen from the figure, the usage of the loop divider as the BIST Controller introduces the need for a new multiplexer to select between the feedback signal in normal PLL operation and the test clock in 'BIST mode'. However, in view of the fact that the initial multiplexer was removed, this is still a significant reduction in hardware.

The number of external signals needed for BIST operation is also reduced from three, i.e. A, S AND TCK to two, i.e. START BIST (A) and TCK.

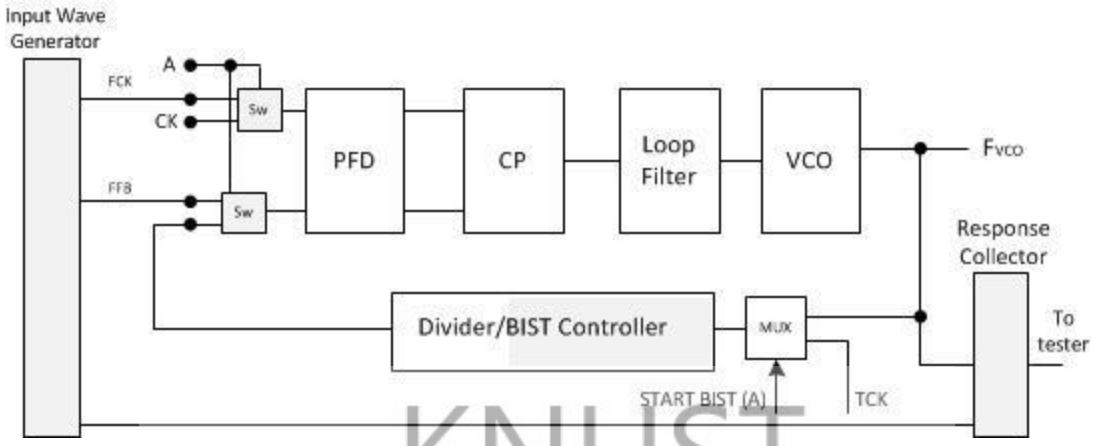


Figure 3.13 - Proposed PLL BIST

3.3.3 Test Pattern Generation

First of all, there is the need for a mechanism which selects between the normal mode of operation of the PLL and the test mode. The use of the 'START BIST' line in Figure 3.13 which is the same signal as 'A' accomplishes this. Zero (0) has been subjectively chosen to start the BIST mode while one (1) selects normal operation mode.

From the proposed BIST scheme and previous knowledge of how PLLs operate it is easy to generate a BIST waveform to implement the charge and discharge test strategy outlined in section 3.3.1. It was established in chapter two that a PLL is a control system which compares a feedback signal to a reference clock signal and adjusts the frequency of the output to meet that of the reference. Hence, as long as the frequency of the feedback signal is less than that of the reference signal, the PLL will be in a continuous charge state. Similarly, as long as the frequency of the feedback signal remains higher than that of the reference signal, the PLL will be in a continuous discharge state.

Based on this concept, test stimuli are designed for the two inputs of the PFD of the PLL which will cause the PLL to charge for a fixed amount of time and then discharge for a certain amount of time. Figure 3.14 shows the BIST waveforms. In the figure TCK and TFB refer to the 'test clock signal' and 'test feedback signal' respectively.

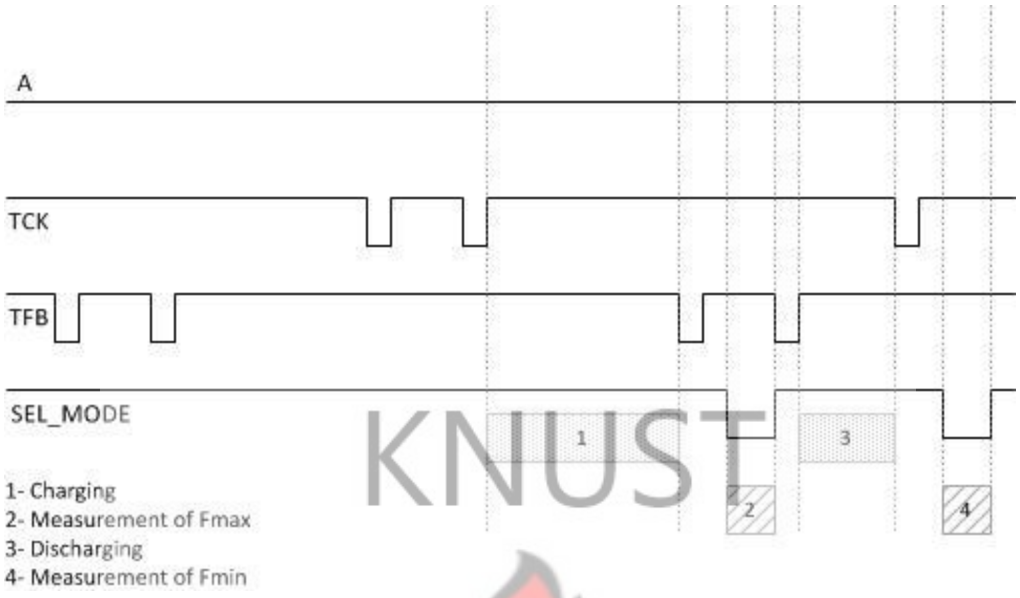


Figure 3.14 - Test Input Waveforms (BIST)

The operation of signals START BIST, TCK, and TFB have previously been covered. From Figure 3.14 though, a new signal SEL_MODE is seen. It alternates between two levels and causes the response analyzer to shift between two states. Its use will become more apparent in the following section.

3.3.4 BIST Hardware

With reference to Figure 3.13, the schematics for the BIST scheme will now be developed.

Switches

Normally, a multiplexer should be convenient for a switching mechanism as in the case of Sw1 and Sw2. However the Figure 3.15 shows an even simpler circuit that meets our requirements.

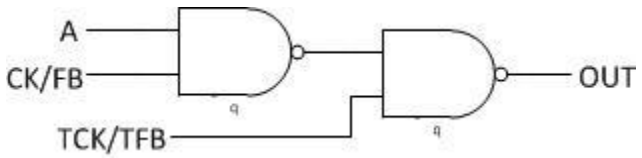


Figure 3.15 - BIST switch schematic

Its operation can be deduced from the truth table below.

Table 3.4 - Truth table for BIST switches

START_BIST	CK/FB	TCK/TFB	OUT
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

From the truth table, if START_BIST = 1 and TCK (or TFB) =1, then the output will always be equal to CK (or FB). On the other hand if START_BIST = 0, then the output will always be equal to the inverse of TCK (or TFB) irrespective of the value of CK (or FB). This might be used as a switching mechanism between CK/FB and TCK/TFB, and reduces the number of logic gates needed as opposed to a traditional multiplexer. The downside to this mechanism is that it can only be used if it is known that the circuit will never go into the sixth state (from the table). Using this kind of switching mechanism also implies that TCK/TFB will always be supplied with the inverse of the actual value needed.

BIST Controller and Multiplexer

Since the loop divider of the PLL is being used as the BIST Controller, the hardware obviously remains unchanged from that presented in section 3.2.5 with the exception that one more flip-flop stage is added as the BIST Controller requires six states.

For the multiplexer, the state of the art design shown in Figure 3.16 is used.

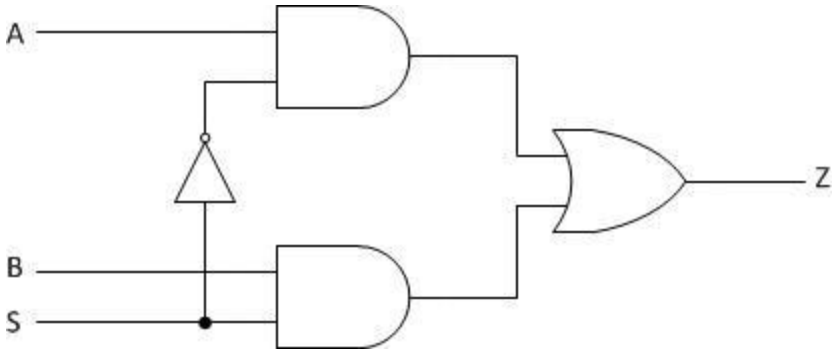


Figure 3.16 - Classical multiplexer design

Input Wave Generator (IWG)

A 6.25MHz clock is used during the BIST to generate the controller's outputs. These outputs are then fed to the combinational circuit which generates the test patterns. The main design question here is this: How is the input wave generator circuit designed? With reference first to Figure 3.14 and later Figure 3.17, the answer becomes obvious.

A 6.25MHz clock will have a period of $^{1}/_{6.25} = 0.16\mu s$. As shown in Figure 3.17, the BIST is run for 10.24µs. This means a BIST controller that is capable of generating $^{10.24}/_{0.16} = 64$ unique states is needed; one state per clock cycle of the BIST clock. A six stage (i.e. $2^6 = 64$) asynchronous counter has just this functionality; hence, the need for an additional stage to the loop divider as stated previously.

Figure 3.17 shows the first half of the *Spectre®* simulation results of the output generated by the BIST Controller without the input clock. Two markers are used to show the states 111111 and 0111111.

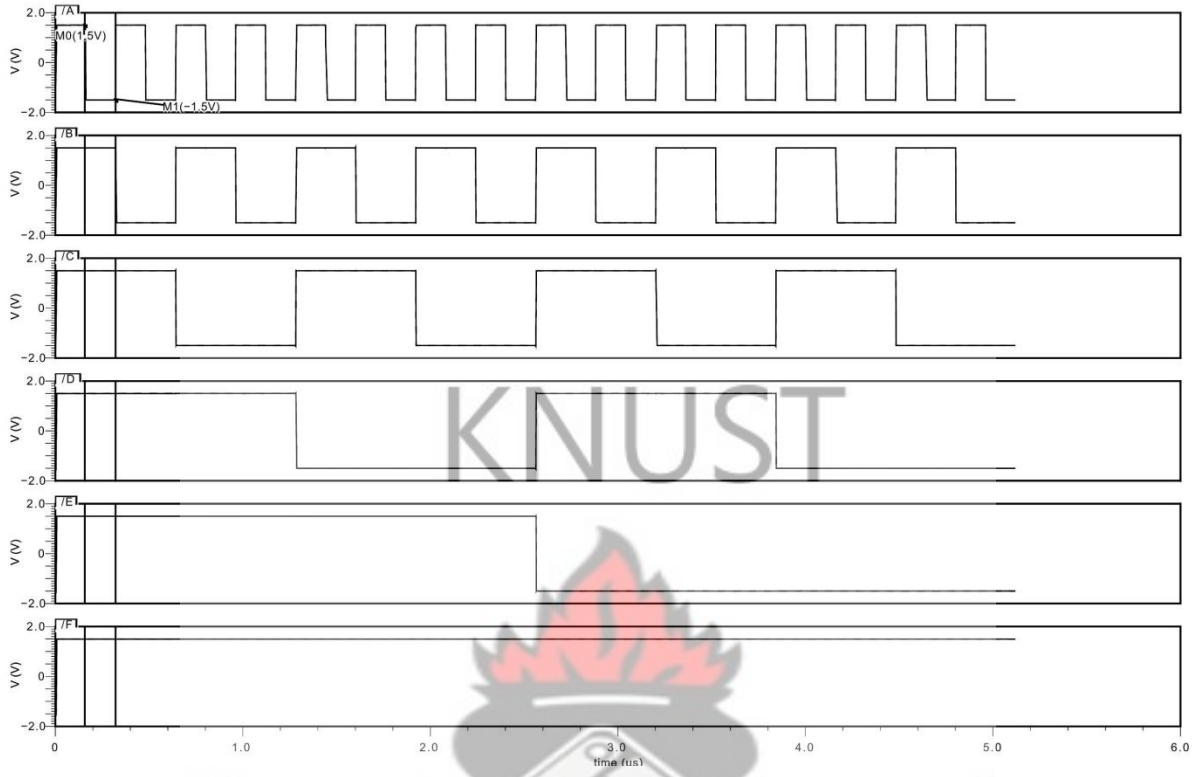


Figure 3.17 - BIST Controller output plot

Restated, by considering Figure 3.17 together with Figure 3.14, taking the timing of the signals into account, a 64-state truth table which uses the outputs of the controller as inputs and signals TCK, TFB, and SEL_MODE as outputs can be generated. The truth table is shown in the appendix. The START BIST signal is not considered since it is tied to logic zero throughout the entire BIST run. From the truth table the boolean functions mapping TCK, TFB, and SEL_MODE to the outputs of the controller circuit can be found, from which the input wave generator circuit can be designed. The following boolean functions are determined.

$$TCK = AB\bar{D}\bar{E}F + AB\bar{C}\bar{D}\bar{E}\bar{F}$$

$$TFB = AB\bar{D}\bar{E}\bar{F} + A\bar{B}DEF$$

$$SEL\_MODE = \bar{C}D + C\bar{D} + B + E + F$$

Based on these boolean functions, the schematic for the input wave generator is shown in Figure 3.18.

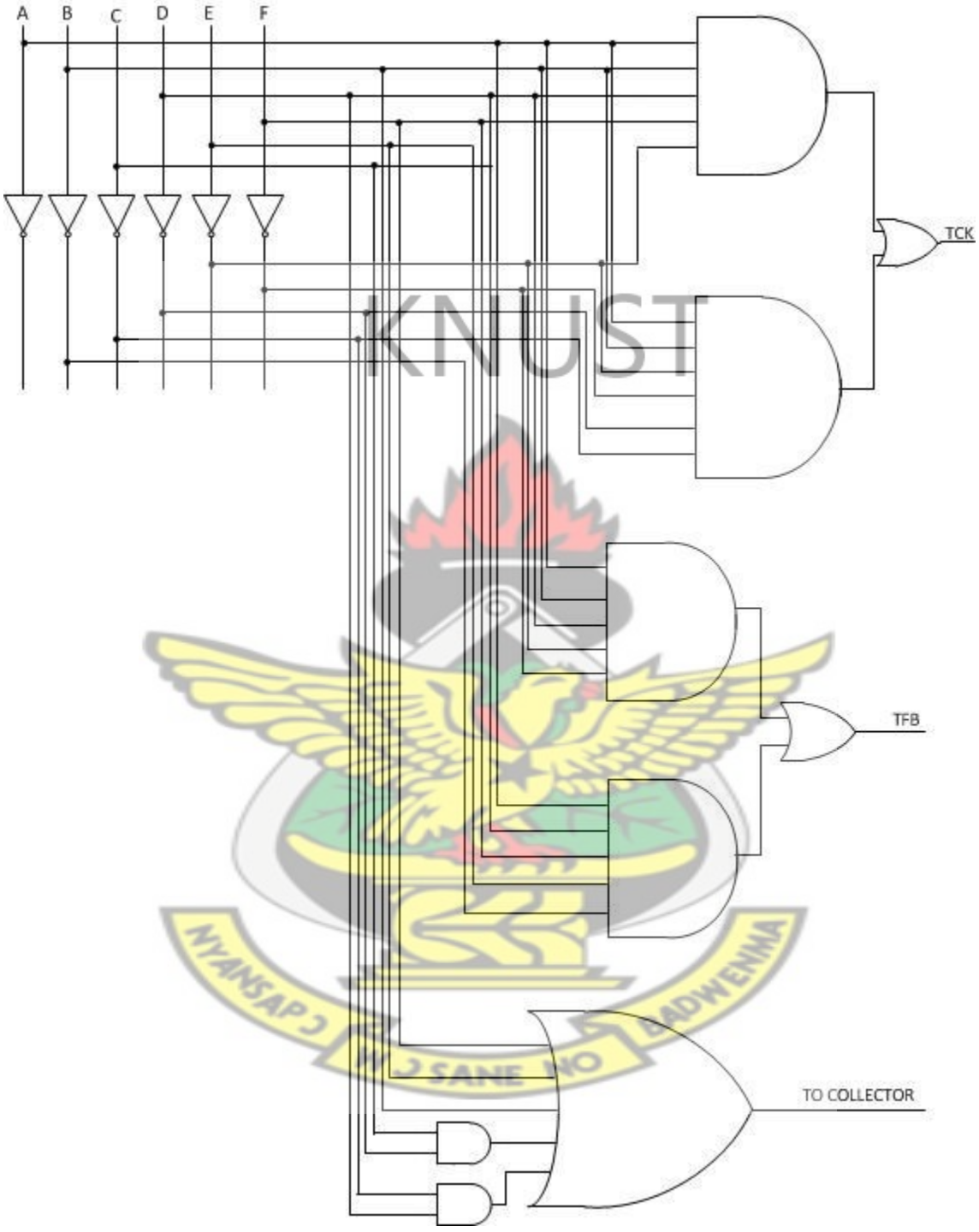


Figure 3.18 - Input Wave Generator circuit

Response Collector

As explained in section 2.7.4, traditional response collectors or analyzers make use of compaction. A response sequence is generated by subjecting a fully working CUT to a series of tests or through computer simulation. The generated sequence is then compacted based on a particular compaction function and stored. During BIST, the actual response is again compacted and compared with the stored pattern. One of the main issues with this approach is that compaction is not a lossless process, and is therefore subject to aliasing [31]. However it still remains the most practical solution for high frequency applications.

For low frequency applications though, it is possible to use an ordinary counter as the response collector. It will suffice to use such a counter in our design for the measurement of F_{max} and F_{min}. After the frequency measurements though, the counted value needs to be shifted out and again, an ordinary shift register may be used for this purpose.

To reduce the hardware cost of using both a counter and a shift register, a simple circuit that can work as both is introduced here (Figure 3.17). The total number of flip-flop stages is n, and it can have any integer value. The circuit is however practical only for small values of n. In this design n has a value of 6.

The operation of the 'counter cum shift register' circuit is quite simple to understand. It employs the inherent ability of flip-flops to be configured to work as counters and as shift registers. Both configurations are built into one system and 'separated' by the use of multiplexers. All multiplexers are connected to one select line, SEL_MODE, whose logic value determines whether the circuit is being used as a counter or as a shift register. A logic zero configures the circuit to work as a counter while a logic one configures it to work as a shift register. This choice is however purely subjective and the converse should work just as well.

When SEL_MODE is zero, the flip-flops are connected through the wires labeled C (i.e. for counter), and the system behaves as the 6-bit binary counter circuit shown in Figure 3.20.

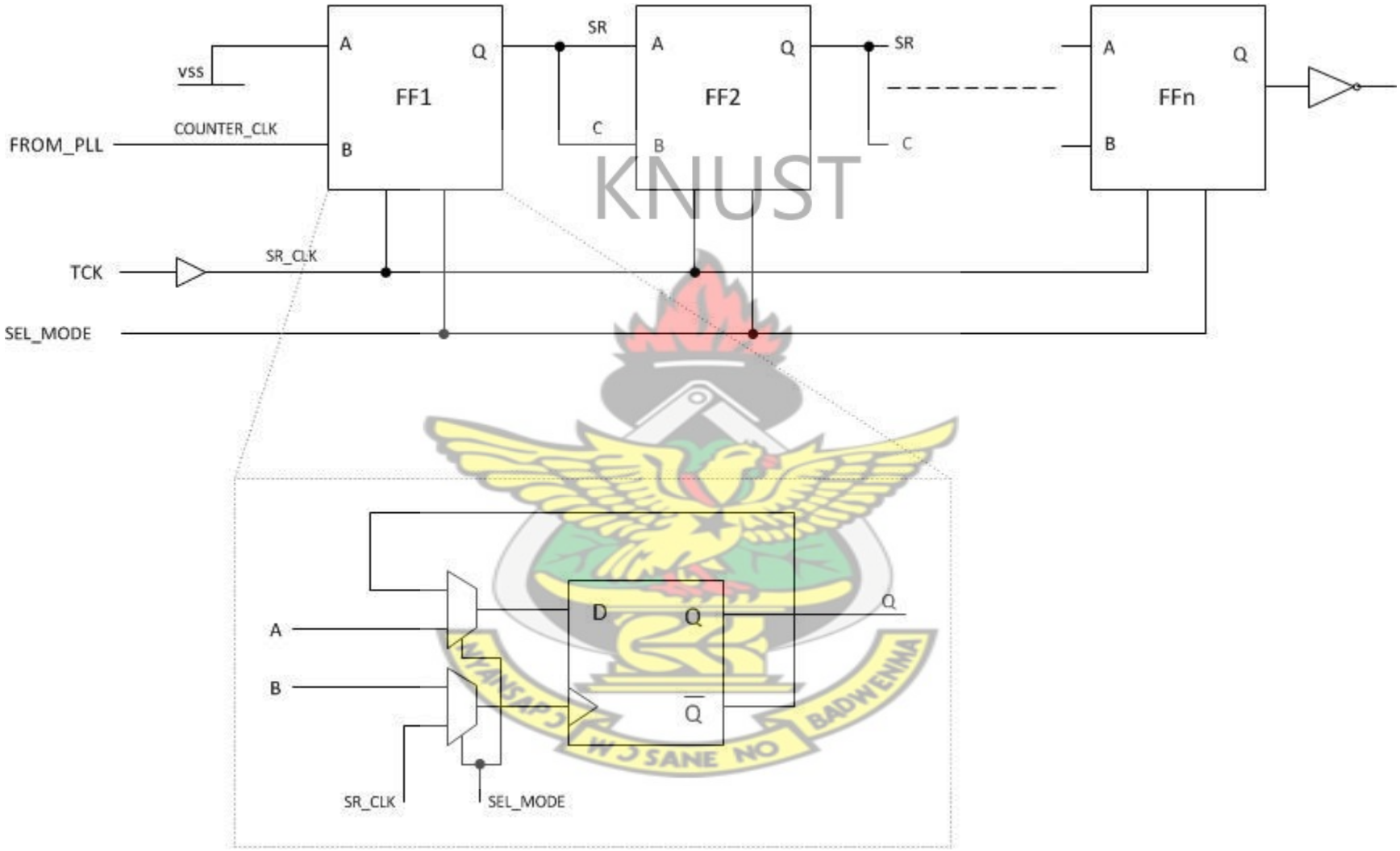


Figure 3.19 - Counter cum shift register design

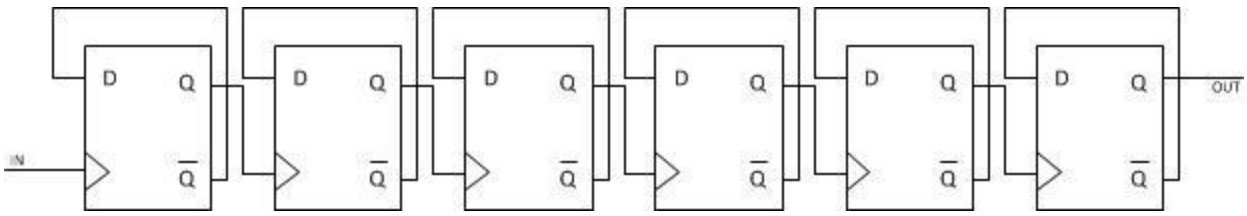


Figure 3.20 - 6-bit binary counter

When SEL_MODE is one, the flip-flops are connected through the wires labeled SR (i.e. for shift register), and the system behaves as the shift register circuit shown below.

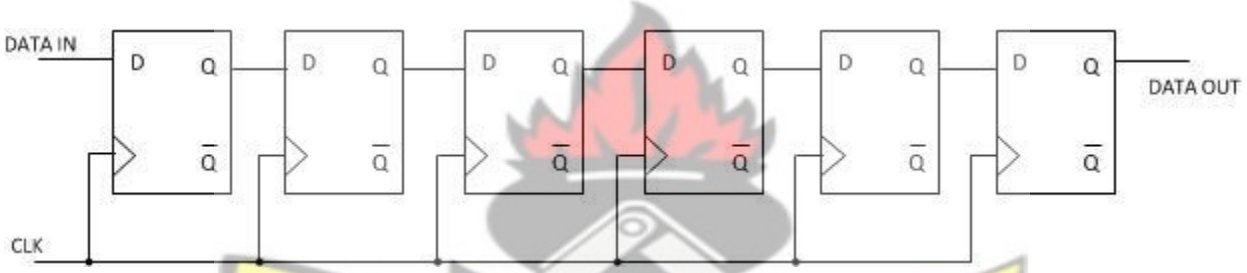


Figure 3.21 - 6-bit shift register

During the BIST run, our response collector is configured mostly to act as a shift register. Tying the shifter's data input to VSS will therefore reset all flip-flops to logic zero. During the frequency measurements, the circuit acts as a counter for the predetermined period of 0.32µs after which it switches back to 'shift mode' to serially shift out the result of the count.
Because a down counter is used, an inverter is added to the last stage causing the value that is shifted out to always be the result of an up count.

A problem arises at the point where the collector is switched from 'count mode' to 'shift mode'. Due to the fact that the SEL_MODE signal is clocked from the same source as the shift register's clock, the rising edge of the two signals coincide. At that instant, the circuit is expected to behave as a shift register, but SEL_MODE is yet to be fully asserted. The result is that for the first clock cycle of the shift register's operation, it fails to function properly, generating unpredictable results. It resumes proper operation as a shift register on the rising edge of the

next clock cycle by which point SEL_MODE will long have been asserted. However at that point, the exact data that needs to be shifted out from the counter will have been altered.

To solve this problem, a delay is inserted between the external clock and the shift registers clock input. This prevents the rising edge of SEL_MODE and the shifter's clock from coinciding. The delay may either be implemented with a series of inverters or a resistor. A resistor is used in this case because it occupies less area in the final layout of the circuit.

The main advantage of using such a system for the response collector is that, since it is lossless, it has the potential for wider fault coverage. It is also simpler compared to a compacter and easier to implement. At first glance, one might conclude that a possible disadvantage of such a collector is that the number of flip-flops needed will scale rather quickly with increasing frequency. It should however be noted that as the frequency increases, the time window within which frequency measurements are taken will decrease accordingly. The number of flip-flop stages required will therefore not necessarily increase with increasing frequency. Hence the proposed response collector design remains practical even for high frequency applications.

3.3.5 The PLL Frequency Synthesizer with BIST

Combining the PLL frequency synthesizer presented in section 3.2.6 with the BIST circuit just developed, Figure 3.22 shows the final design of the 40-100MHz phase-locked Loop frequency synthesizer with built-in self-test.

Summary

In this chapter, a PLL based frequency synthesizer was designed. A new BIST scheme was proposed and its implementing hardware was developed. In the next chapter, the PLL frequency synthesizer designed in this section will be tested and simulated on various levels to verify its functionality. Similar simulations will be carried out to verify the BIST scheme.

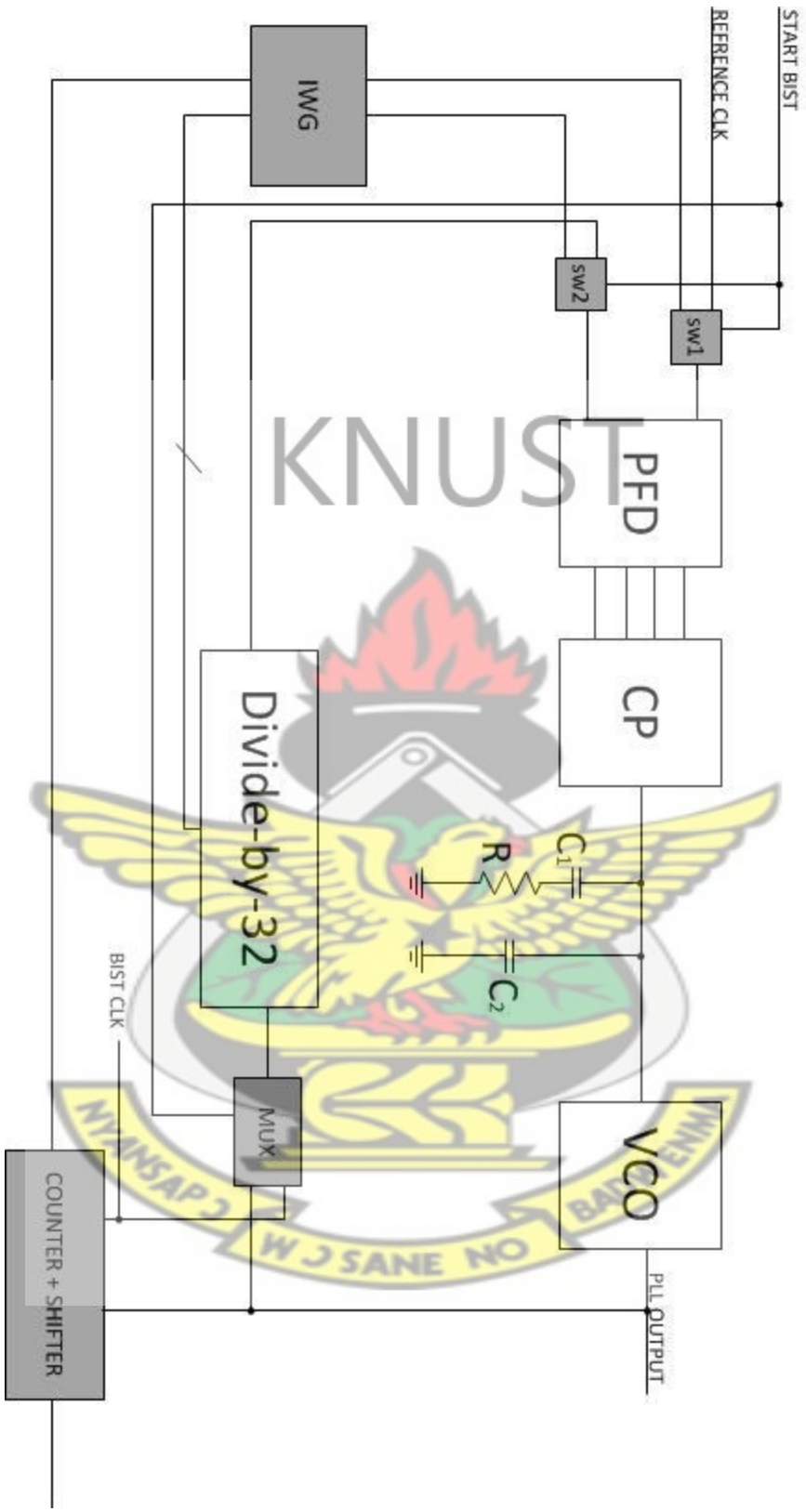


Figure 3.22 - Designed PLL frequency synthesizer with BIST

Chapter 4

Experimental Results

4.0 Introduction

In this chapter all building blocks that make up both the PLL and BIST scheme are verified. The results are compared to our initial expectations. Where applicable, the obtained results are compared to existing work. The entire design is also simulated on a system level to verify its operation. Almost all simulation results are presented in a graphical format. Simulations are carried out using the *Cadence® Virtuoso® Spectre® circuit simulator*.

4.1 Voltage Controlled Oscillator

The first block that is checked for correctness is the VCO. The primary function of a VCO is for the output to oscillate at different frequencies as the control voltage is changed. Figure 4.1 shows a parametric analysis of the VCO output voltage over different control voltages.

A few other VCO characteristics or parameters are measured and evaluated under varying conditions. These include linear sensitivity, phase noise, and power dissipation. These characteristics determine the overall characteristics of the PLL.

Tuning Sensitivity

Tuning sensitivity is defined as the frequency change per unit of tuning voltage. Ideally tuning sensitivity would be constant but in practice this is generally not the case [41]. To compute tuning sensitivity, the VCO frequency for different tuning voltages is calculated and the frequency measurements are plotted against the tuning voltage. The gradient of the curve

obtained is the tuning sensitivity and it may be evaluated for different control voltages. The tuning sensitivity is expressed in Hz/V. A plot of VCO output frequency versus tuning voltage is shown in Figure 4.2. From this characteristic, the sensitivity of the VCO between 0 and 0.2v is found.

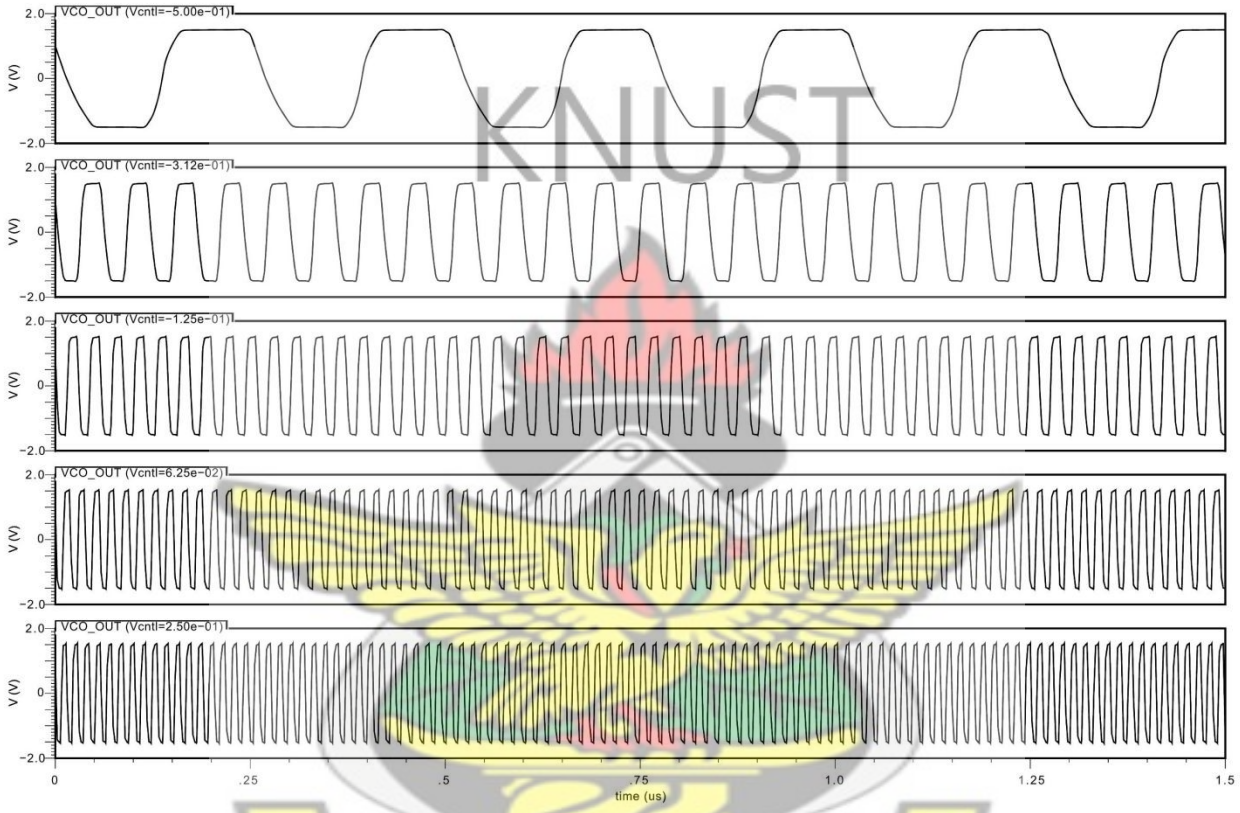


Figure 4.1 - VCO output waveform for different control voltages

$linear\ sensitivity$	$= \frac{\frac{64.89 - 47.15}{0.2 - 0.0} + \frac{64.89 - 53.09}{0.2 - .06443}}{2}$ $= 87.87 MHz/V$ at 0.2v

Phase Noise

Oscillator phase noise is random phase variation in the VCO's output oscillating signal and represents the short term stability of the oscillator [41]. It is generated mainly by the active devices, such as the transistors. Phase noise is the ratio of the output power divided by the noise power at a specified value and is expressed in dBc/Hz. The main source of noise in the PLL is the oscillator and phase noise is the most significant source of noise in oscillators, which makes it a crucial measurement.

Figure 4.3 shows the VCO phase noise at various reference frequencies. Any of these frequencies may be chosen as the reference point for a measure of the VCO phase noise. Choosing the very first point as seen from the figure, the phase noise is read as -71dBc/Hz at 1kHz.

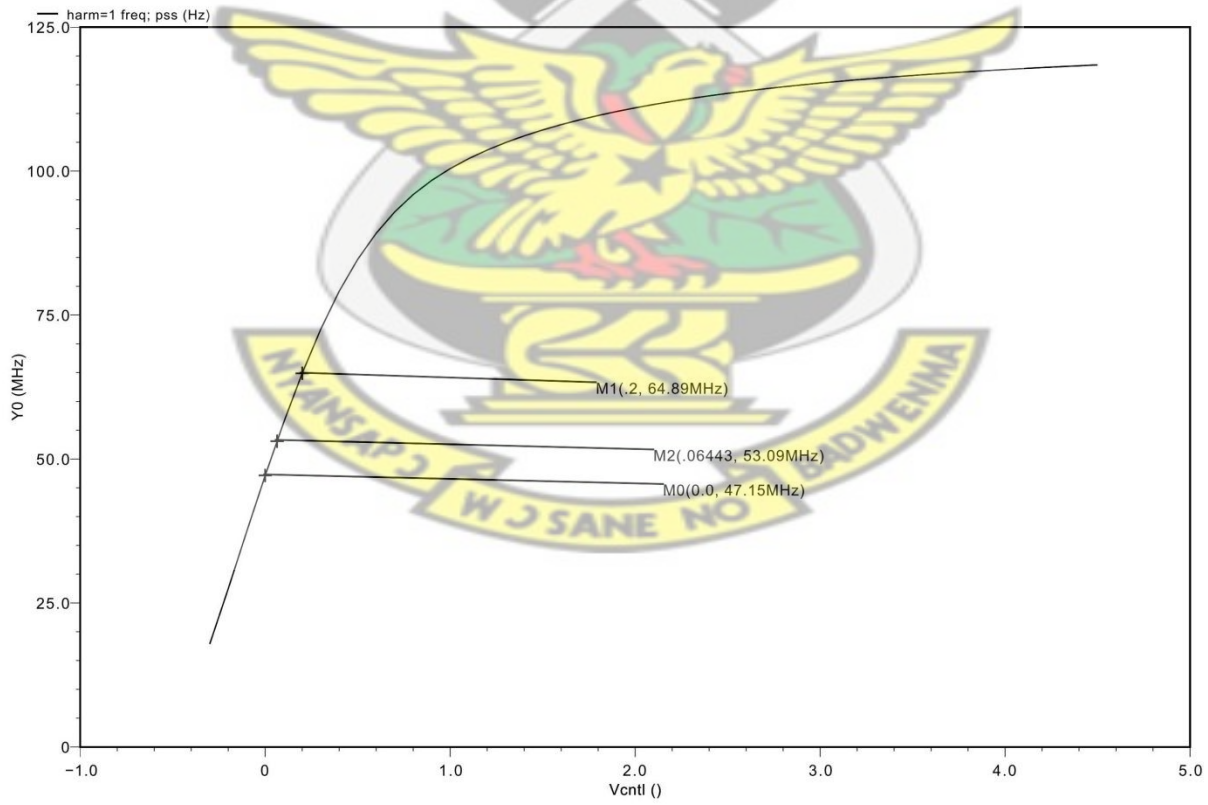


Figure 4.2 - VCO output frequency vs tuning voltage

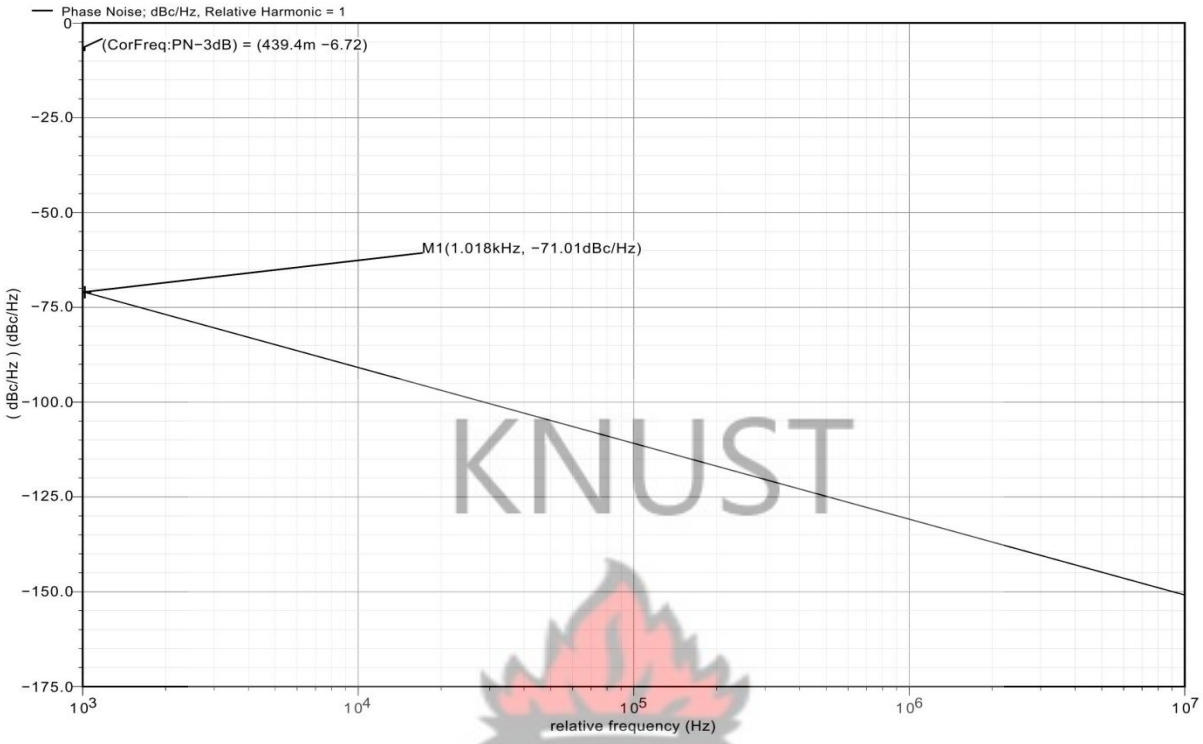


Figure 4.3 - VCO phase noise measurement

Power Dissipation

In VCOs power may be consumed due to switching current from charging and discharging parasitic capacitances, as well as leakage current and subthreshold current. Figure 4.4 shows the power dissipation output of the VCO. Power consumption will vary at different output frequencies and is therefore quoted with a particular frequency reference. In the graph, a marker is placed on dissipated power at 0.0 reference frequency to make it easier to read. The power consumption is $2.5558798 \approx 2.556$dBm at 0 MHz frequency. This is equivalent to about 1.803mW of power.

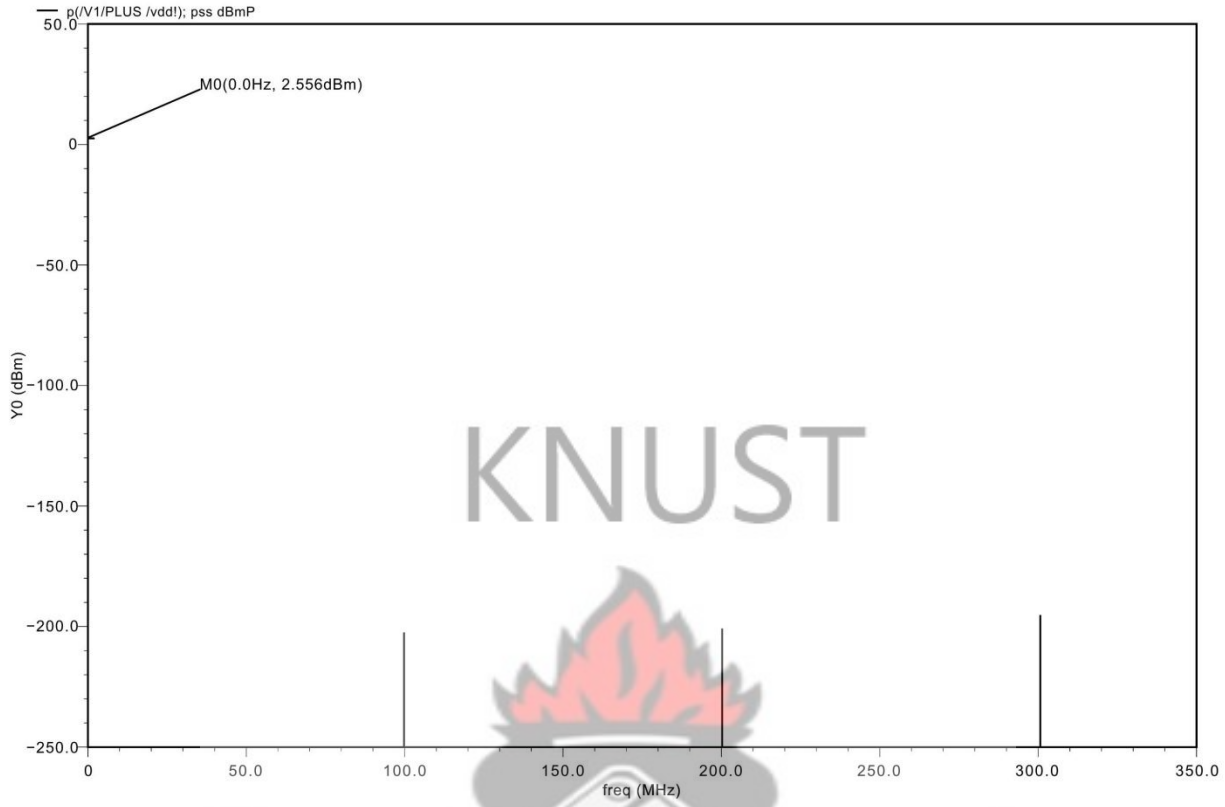


Figure 4.4 - VCO power dissipation plot

4.2 Phase/Frequency Detector Simulation

To test the behavior of the phase/frequency detector circuit, a simple test circuit is used. Two different pulses A (for the reference signal) and B (for the feedback signal) are applied to the two inputs of the PFD.

Three cases are studied:

i. A and B are out of phase, but oscillate at the same frequency

ii. A and B oscillate at different frequencies, but are in phase

iii. A and B are out of phase, and have different frequencies.

For simplicity, only the simulation results of CASE III are presented in the graph in Figure 4.5. The feedback signal, B, has a smaller period and hence a higher frequency than the reference

signal. For the feedback signal's frequency to match that of the reference, it needs to be reduced. As seen in the simulation results, the PFD therefore produces 'DOWN' pulses to the charge pump, which will cause the draining of current from the loop filter, reduction of the control voltage, and subsequent reduction of the VCO's output or feedback signal.

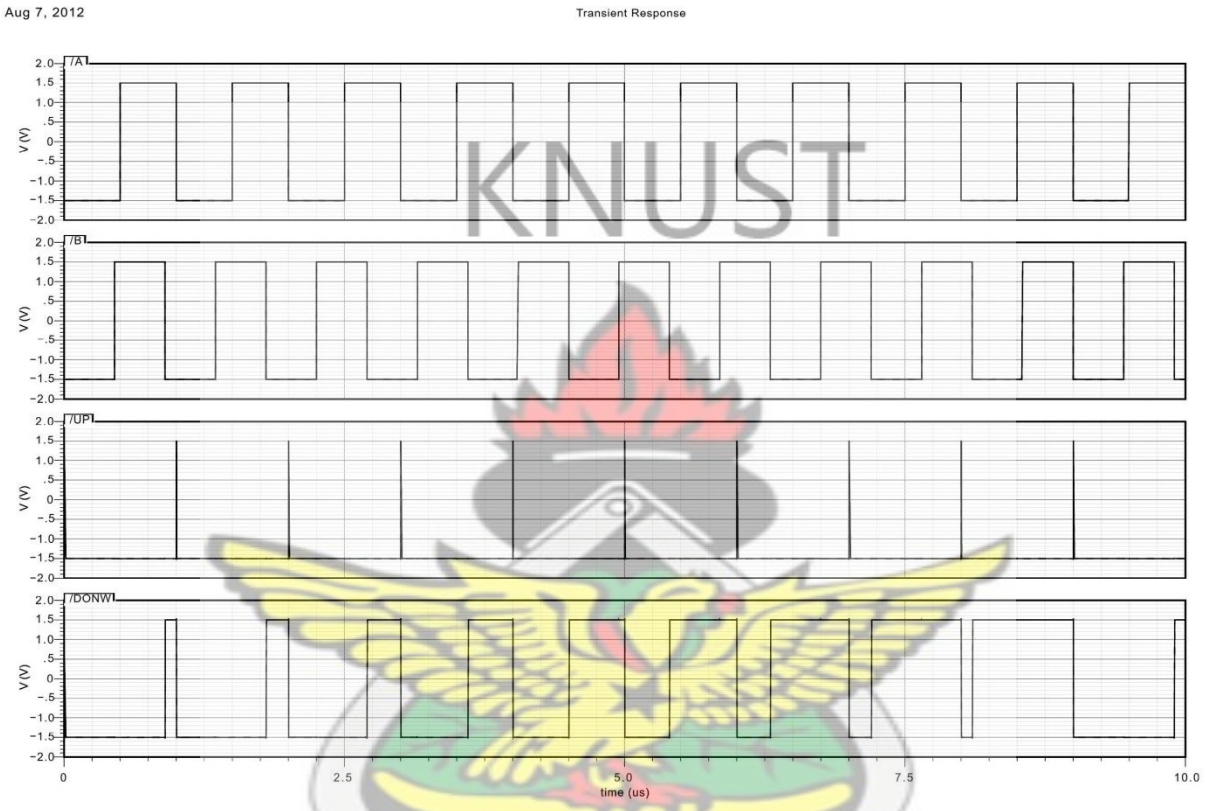


Figure 4.5 - Phase/Frequency detector simulation results

4.3 Charge Pump Simulation

To verify the behaviour of the charge pump, the average output current over a relatively long period of time is measured. This is done for both the charge and discharge phase. The average current that is pumped into or drawn from the filter should be approximately 25µA in both cases. The test circuit used to measure this behaviour is shown below. The test circuit consists of two pulses made up of $\pm 1.5V$ square waves and a large load of 10nf to reduce simulation time. The two square waves are logic complements of each other and alternate between turning on and off the UP and DOWN transistor switches. Thus the expected effect is for the

control voltage generated by the 10nf capacitive load to increase to a limit and decrease. The output is shown in Figure 4.6.

Between the time 0ms and 1.4ms, the UP switch is on, and the control voltage increases gradually from -1.5V to +1.5V. During this interval the charge pump pumps an average current of -24.9µA into the capacitor as determined from CADENCE. Similarly, between the time 2.5ms and 3.8ms, the DOWN switch is on, while the UP switch is off. During this time interval the charge pump sinks an average current of -24.5µA from the capacitor.

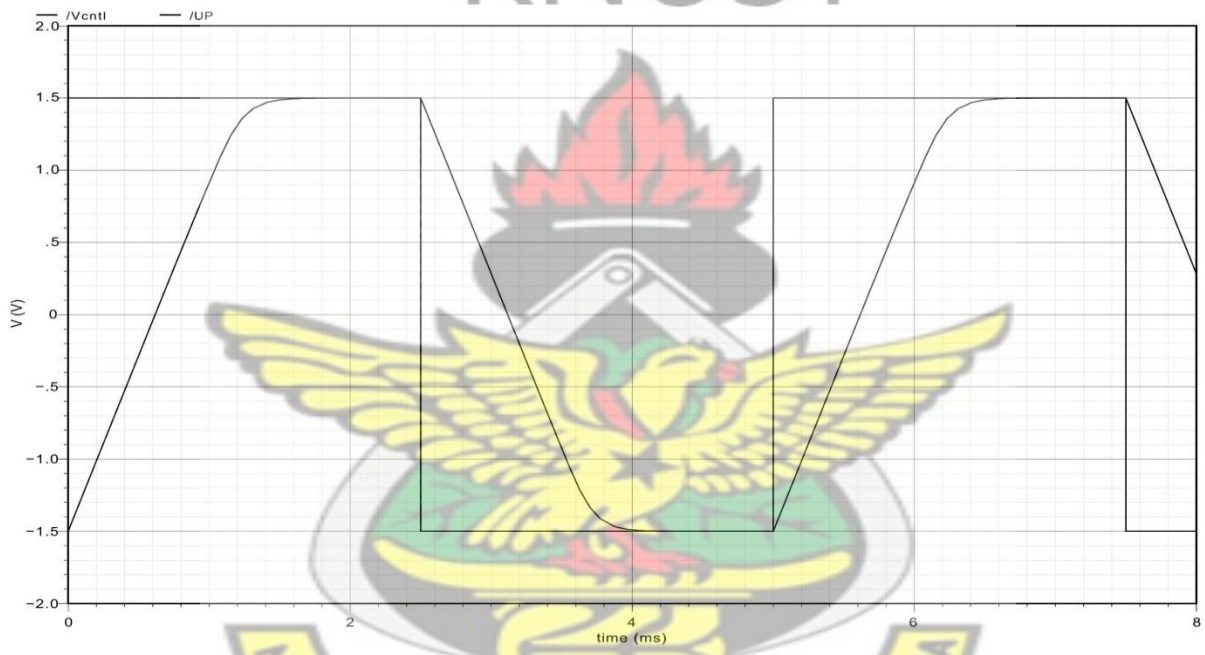


Figure 4.6 - Charge pump output response

4.4 Loop Filter Simulation

The behaviour of the loop filter is best studied through observation of the filter output when the reference clock and feedback signal to the PFD oscillate at different frequencies. This test combines the PFD, charge pump and the filter itself. Figure 4.7 shows the output of this setup when only capacitor c_1 is used as the filter.

The capacitor generates an output voltage that increases periodically as long as the frequency of the feedback signal remains lower than that of the reference.

Figure 4.8 helps us to appreciate this concept better by providing a zoomed in version of the same control voltage shown in Figure 4.7.

Finally, Figure 4.9 shows the same output measured with resistor R_2 included in the loop. The ripple effect mentioned and explained in chapters two and three is seen.

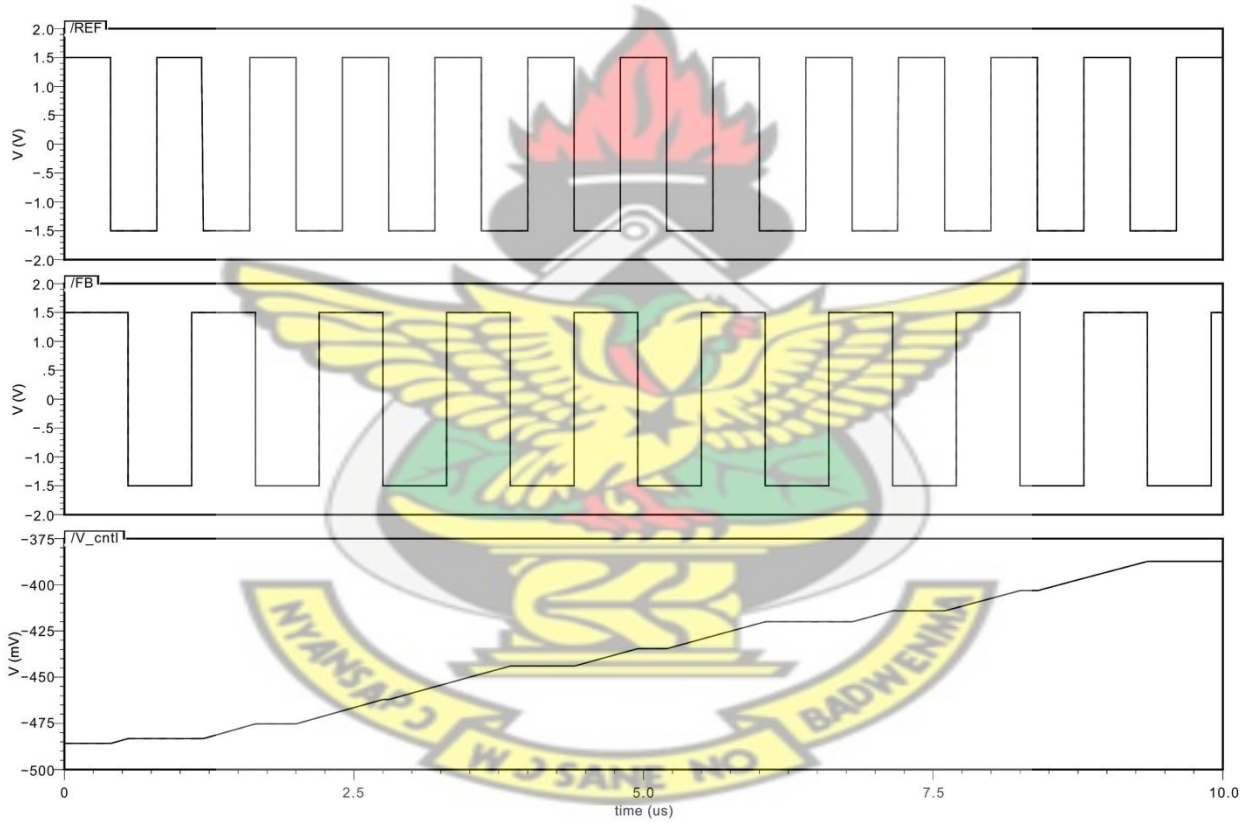


Figure 4.7 - PFD, Charge pump and loop filter interaction

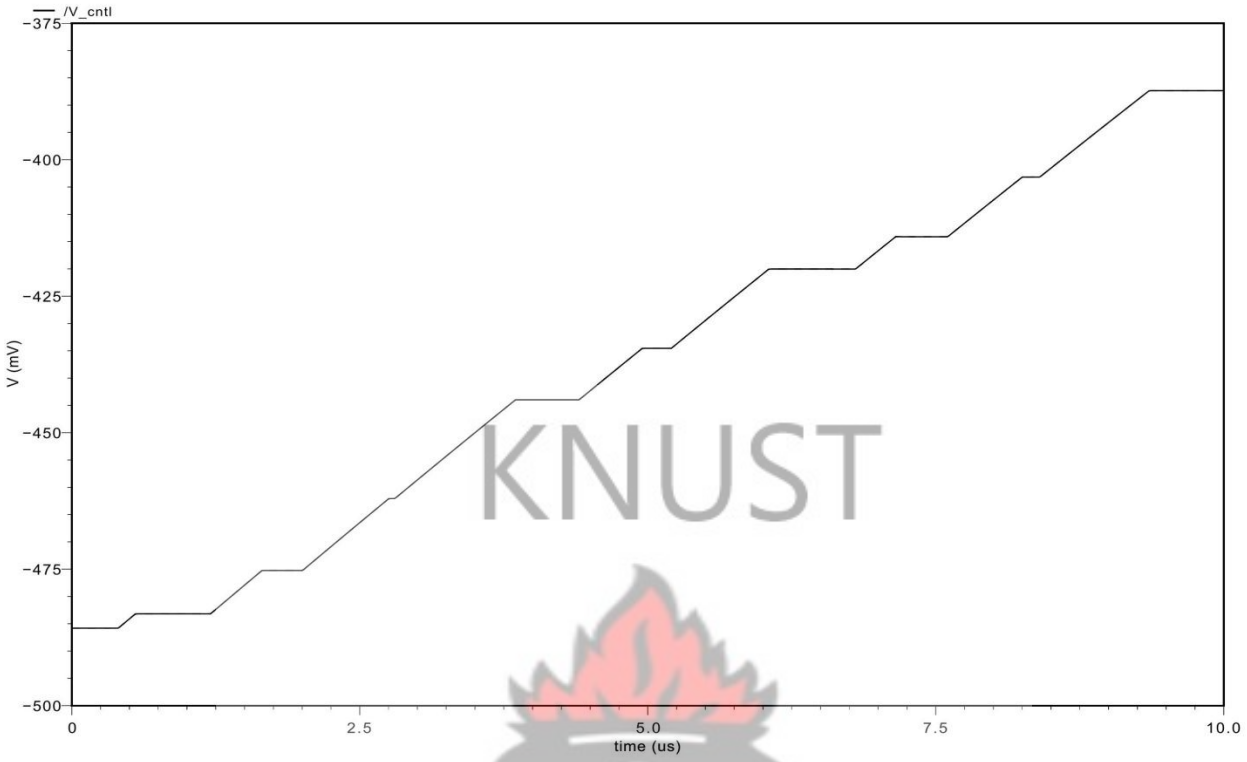


Figure 4.8 - Magnified version of loop filter output

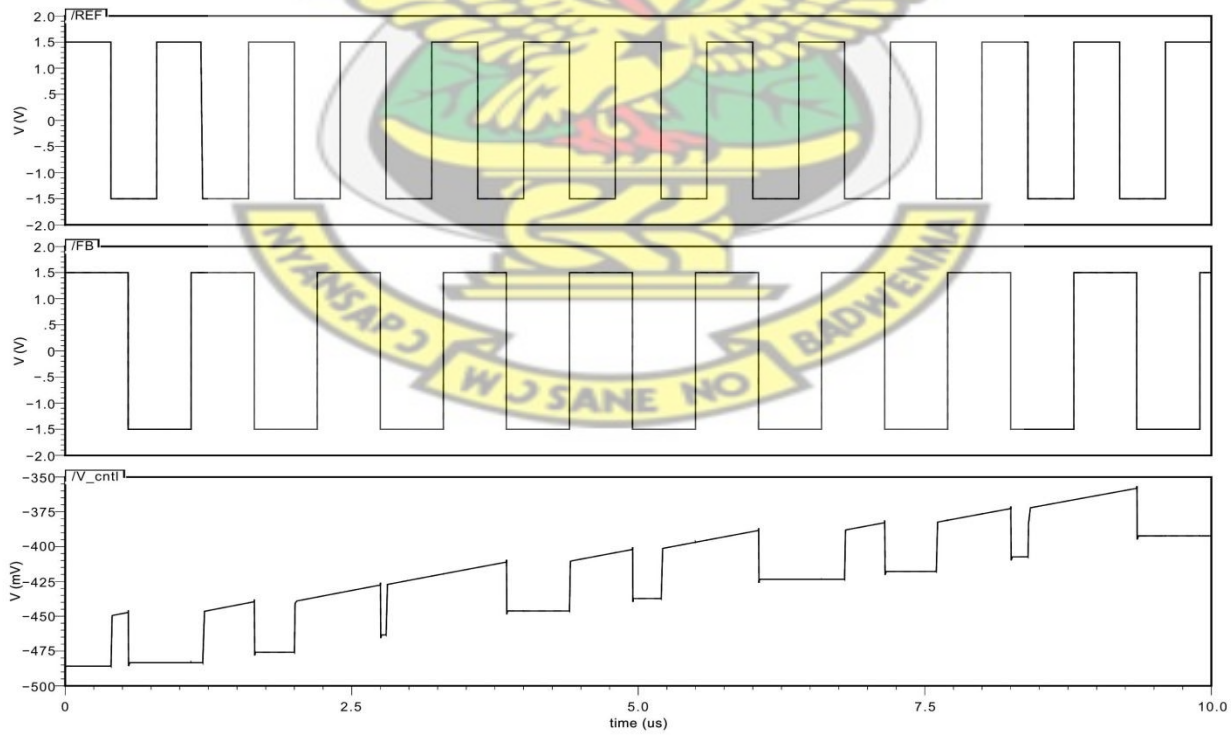


Figure 4.9 - Loop filter output considering R_2

4.5 Frequency Divider Simulation

Simulating the frequency divider's behaviour is a very straightforward process. A pulsating source is applied as an input to the divider and each of the five outputs is observed for frequency division. Figure 4.10 shows the output when the simulation is done using a 10MHz clock input.

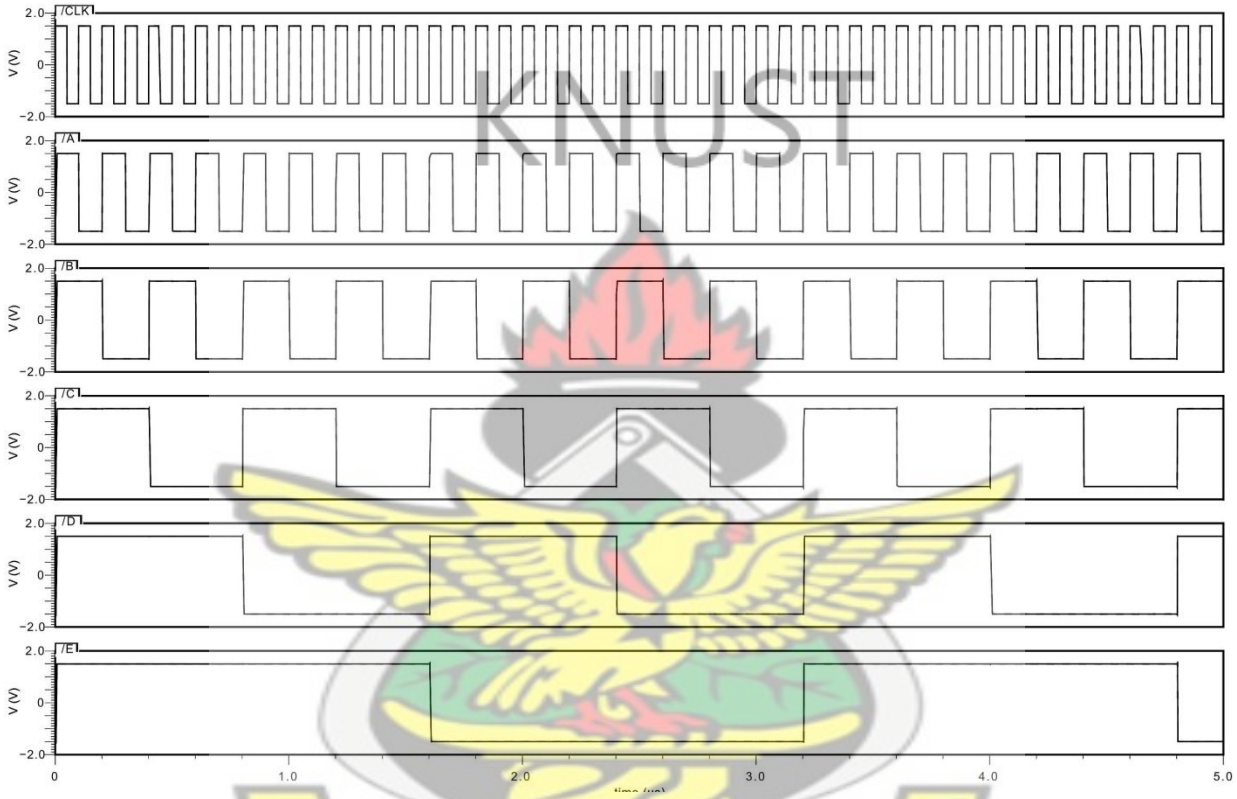


Figure 4.10 - Frequency divider simulation results

4.6 PLL Simulation

The final step in verifying the PLL design for correctness is to simulate the PLL itself. To do this the PLL is used to synthesize five sample frequencies. The output frequency is observed to see if it matches the expected value. The lock time is also observed. Since the designed PLL uses a divide by 32, a signal with a frequency of $^{f}/_{32}$ is needed to synthesize an fMHz signal. The sample frequencies are chosen to cover the operating range of the PLL, i.e. minimum, close to

minimum, mid-band, close to maximum, and maximum frequency. Hence an attempt is made to synthesize 40MHz, 55MHz, 70MHz, 85MHz and 100MHz signals.

Figure 4.12 shows the PLL synthesizing a 40MHz signal using a 1.25MHz clock and an 85MHz signal using a 2.66MHz clock signal as reference.

Figure 4.11 - PLL synthesizing a 40MHz signal

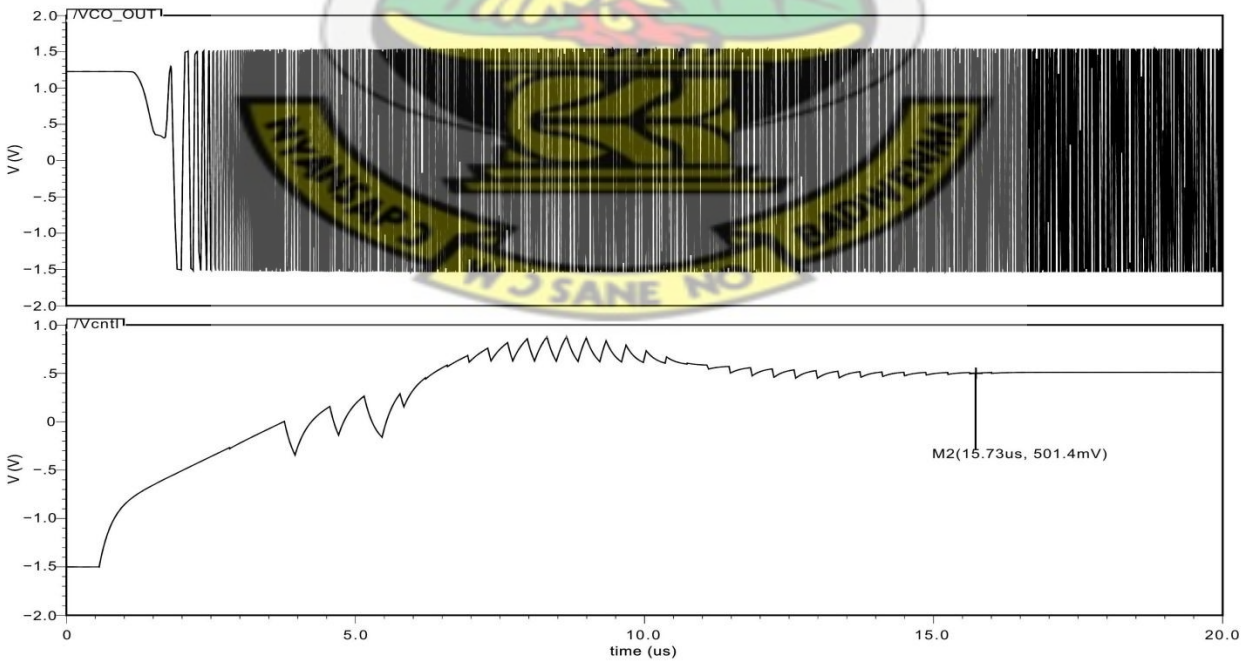


Figure 4.12 - PLL synthesizing a 70MHz signal

In the first case, the PLL, from an initial frequency of just under 2Hz, settles to exactly 40MHz after 11.63µs. Similarly, in all other cases, the PLL is able to acquire the exact frequency being synthesized albeit in different acquisition times. Table 4.1 shows some aspects of the simulation results. All simulations are carried with a starting oscillation frequency of about 2Hz.

Table 4.1 - PLL frequency synthesis comparison

	Reference	Settling Frequency	Acquisition time
(i)	1.25000MHz	40MHz	11.63µs
(ii)	1.71875MHz	55MHz	12.19µs
(iii)	2.18750MHz	70MHz	10.93µs
(iv)	2.65625MHz	85MHz	15.73µs
(v)	3.12500MHz	100MHz	25.12µs

4.7 BIST Simulation

BIST simulation involves simulation of the BIST hardware and fault simulation. The major BIST hardware simulated are the input wave generator and response collector. The operation of the controller is not shown here because it is exactly the same as the loop divider.

Input Wave Generator Simulation

The input wave generator is simulated to verify its operation. The output waveform is compared to the test pattern designed in section 3.3.3. Simulation is carried out by connecting a pulse source with a period of 0.16µs to the BIST controller which is connected directly to the generator, and observing its (i.e. the wave generator's) output for 10.24µs. The changes in the control voltage to the VCO and the resulting effect on the output frequency of the PLL are studied alongside the test pattern.

From Figure 4.13, with the exception of a few glitches, the test pattern obtained is as expected (see Figure 3.14). The glitches are however not a point of major concern because they do not affect the BIST. It is also noted that the signals for TCK and TFB are inverted. Considering the inverting nature of switches Sw1 and Sw2 though, this is exactly the intended nature of the

design.

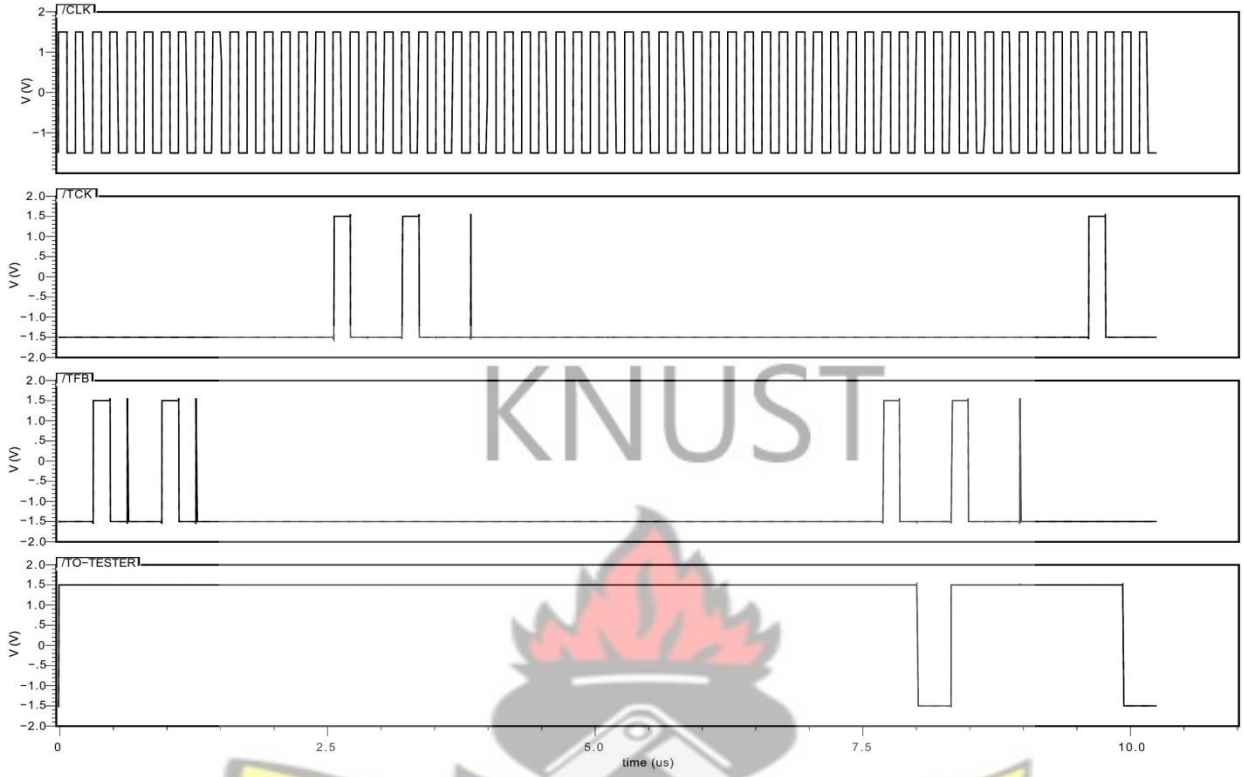


Figure 4.13 - BIST waveform generator simulation output

Response Collector Simulation

The response collector circuit is simulated to verify the counter cum shift register action using a three stage version of the design. A pulse source alternating between -1.5v and 1.5v (i.e. logic 0 and 1) is used. The counter clock is supplied with an arbitrary 10MHz signal. The pulse source is configured to count (set to logic 0) for the first 640ns and shift (set to logic 1) for the next 640ns. The simulation is run for 1.28µs.

Figure 4.14 shows the simulation results. Between the start and the point M0, shown with the vertical marker, SEL_MODE is zero and the analyzer counts. Six cycles of the 10MHz signal in total are 'down counted' thus generating the value 010. However at the point of switch between counting and shifting, the signal being counted is caught somewhere within the seventh count. As SEL_MODE and the counter clock are not perfectly synchronized this is a situation that will occur a large percentage of the time. Hence even though the seventh count is

incomplete, the value that is shifted out is still 001. In the final analysis a ± 2 error margin is used to account for such variations. In the figure, shifting starts from the marker M0 to the end. Since a three-stage collector is used however, only three stages of the shifting process are considered. Finally, the value that is read from the test pin is considered one clock cycle before shifting since the last stage of the counter is directly connected to the output pin and will automatically hold the first shift value of interest.

Figure 4.15 shows the final shifted output read from the BIST output pin as 110 as expected.

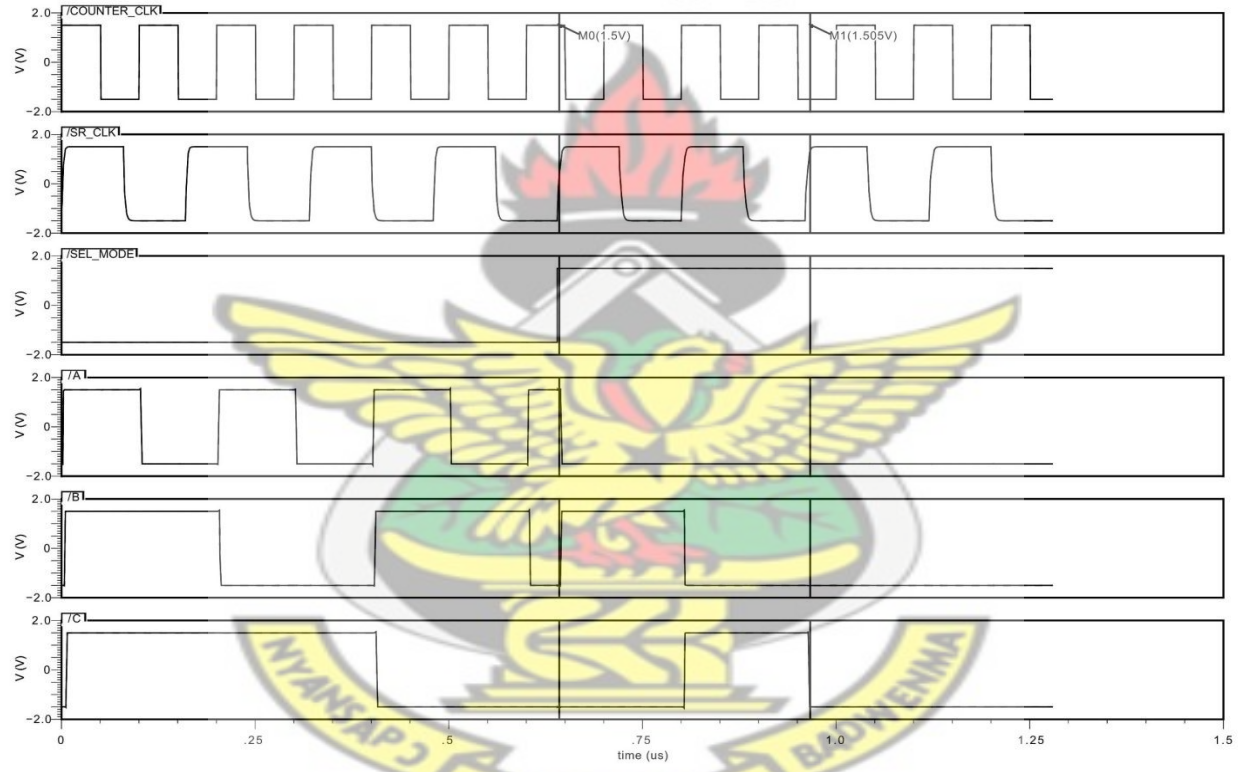


Figure 4.14 - Three stage response collector simulation

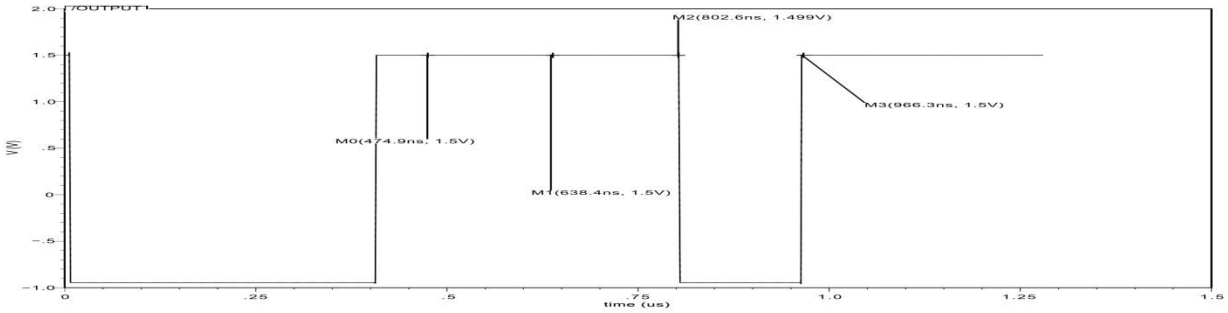


Figure 4.15 - Final output read from the three stage response collector

BIST circuit simulation

The behaviour of the entire BIST circuit is finally observed from simulated output. The testing phase is demonstrated in the simulation shown in Figure 4.16. During the charge phase, the PLL charges from a very low frequency to a maximum frequency of about 84MHz. There is then a hold state of 0.48µs. This is when the measurement of f_{max} is taken. The PLL then discharges to a minimum frequency of about 74MHz. f_{min} is read during the following hold state.

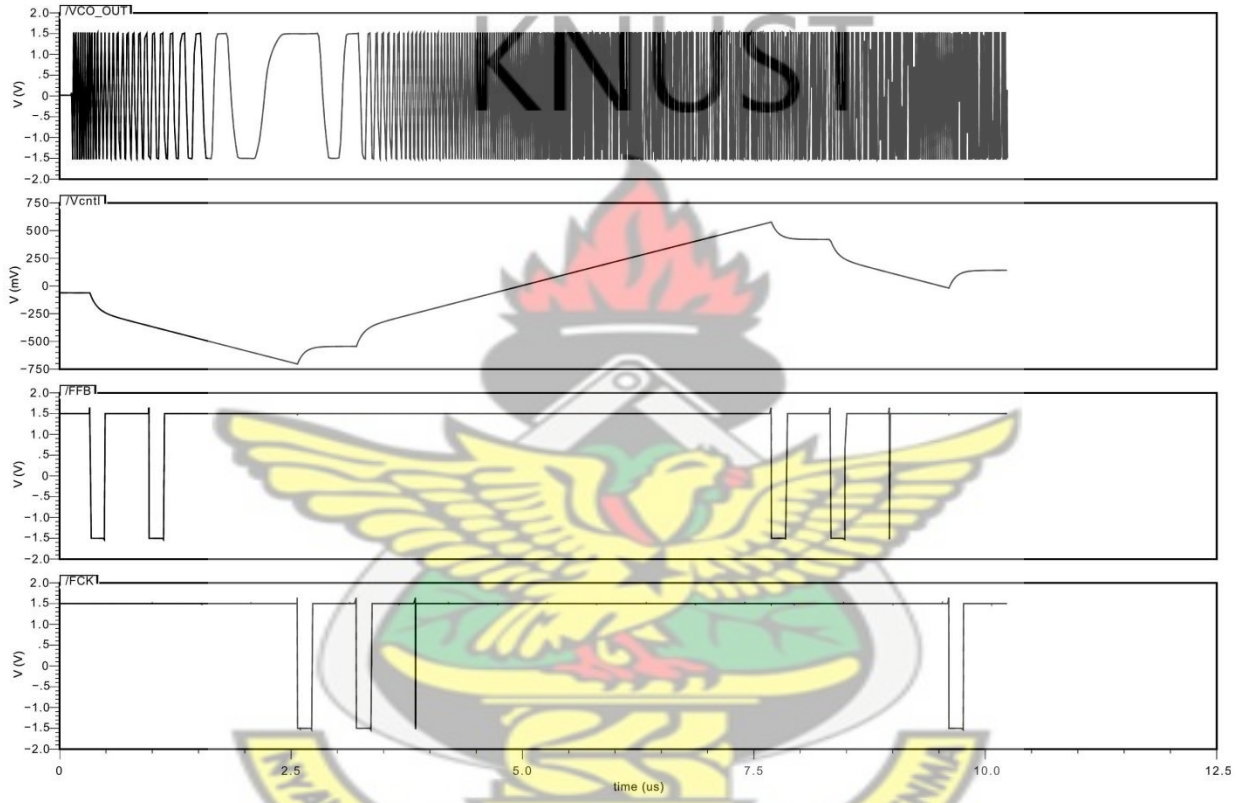


Figure 4.16 - BIST scheme

Table 3.1 compares results for different runs to measure both F_{max} and F_{min}. The results suggest that for both measurements, the same value is always obtained for a working PLL CUT. The value obtained for F_{max} is 011010 which is a decimal value of 26. Since counting is done in a 0.32µs window, this corresponds to a frequency of $^{26}/_{.32} = 81.25\text{MHz}$. Similarly, F_{min}= 23 which corresponds to a frequency of 71.825MHz.

Table 4.2 - BIST output for different simulations

Simulation	F_{max}		F_{min}	
	Binary output	Decimal value	Binary output	Decimal value
1	011010	26	010111	23
2	011010	26	010111	23
3	011010	26	010111	23
4	011010	26	010111	23
5	011010	26	010111	23

It should be mentioned that the simulation environment set up in *Spectre®* is devoid of external factors such as variations in temperature which in practice could introduce slight variations in the measured values.

The BIST scheme presented offers very wide fault coverage. In [11], 316 faults were simulated using *Spectre®*. Out of the 316 faults, only 8 faults, which were later found to be redundant, were undetected. The implemented BIST was also found to save 500µs of the time it takes to test a PLL using the industrial standard specification-based method.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

The phase-locked loop is an essential component of modern electronics systems. In this work, a phase-locked loop was designed and implemented as a frequency synthesizer with a tuning range of 40MHz to 100MHz. All the individual building blocks were designed and tested through simulations in *Spectre*® in CADENCE IC®. Good results were obtained for the synthesis of various frequencies within the operating range. In all cases the exact values of the frequencies being synthesized were obtained within reasonable acquisition times. The PLL exhibits phase noise of -71dBc/Hz at 1kHz and consumes 1.803mW of power at a frequency of 0MHz.

Implementation of a BIST scheme based on an existing test strategy was also proposed. The BIST scheme was made as generic as possible to make it easily portable to other PLL designs. BIST hardware overhead was kept minimal, making it practical for integration. For example, the loop divider of the PLL was employed as the BIST controller.

The output of the BIST was designed to be purely digital and read from just a single output pin, making the testing process seamless and the eventual IC easier to use in larger circuits. Fault simulations reveal very high fault coverage for the BIST scheme, with most undetected faults being redundant.

A very simple response collector which makes use of dynamic flip-flops to perform the functions of both a counter and a shift register was also proposed and implemented. This response collector has the advantage of being simpler than the traditional compacters. It is also lossless unlike available compaction algorithms, and is quite practical for integration in both low

and high frequency applications.

Simulation of the BIST hardware shows an exact match between the test pattern generated by the circuit and the test pattern originally outlined for the BIST scheme. The results also reveal that the test circuit is capable of performing the four required tests of checking the start sequence, checking the stop sequence, changing the frequency dynamically, and normal mode operation. Final BIST output readings are found to be highly consistent across different simulation runs albeit without variations in external conditions as temperature.

5.2 Recommendations

The ring architecture was adopted for the PLL's VCO design to keep it simple and also reduce die area in the eventual layout of the circuit. This architecture however meant that the PLL suffered from a relatively higher phase noise of -71dBc/Hz at 1kHz. In future work, the use of a differential ring oscillator will greatly improve the noise performance. This will also result in a wider frequency range PLL making it viable for a wider array of applications.

There are applications in which although physical isolation of individual ICs is possible, it may not be preferable. An example could be systems employed in nuclear reactors and large power plants. One possible solution to this problem might be the integration of wireless technology with built-in self-test. This will make it possible to initiate, stop, as well read the output of the BIST sequence from a relatively remote distance. Future research could investigate this interesting possibility.

Appendix A: Schematics and Layouts

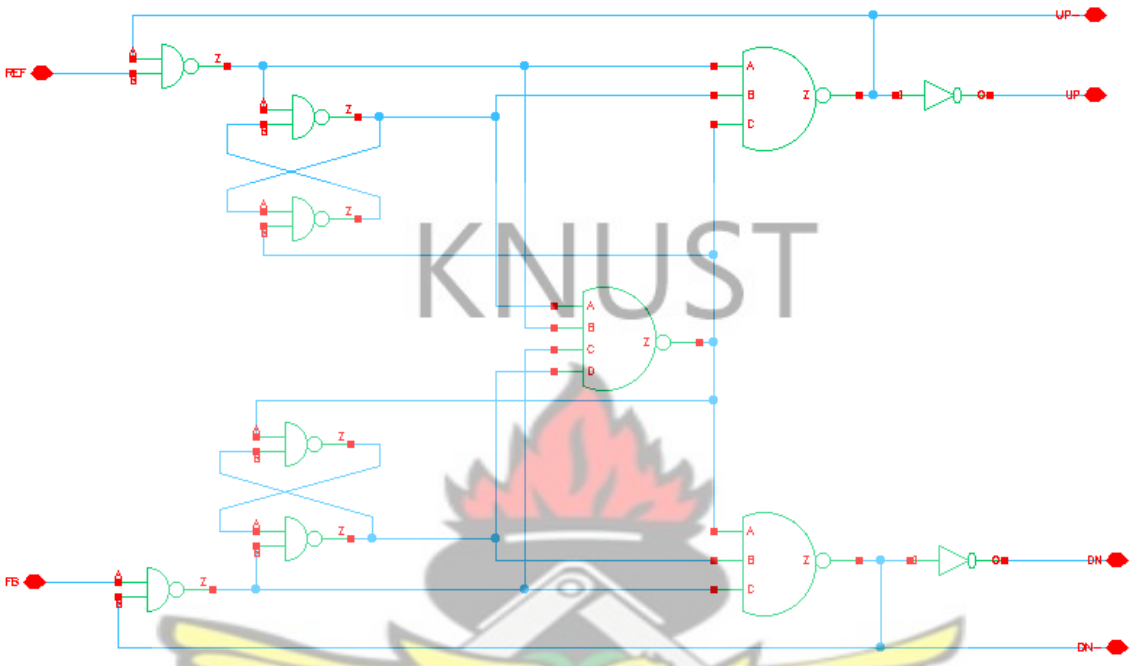


Figure A.1 - Phase/frequency detector schematic

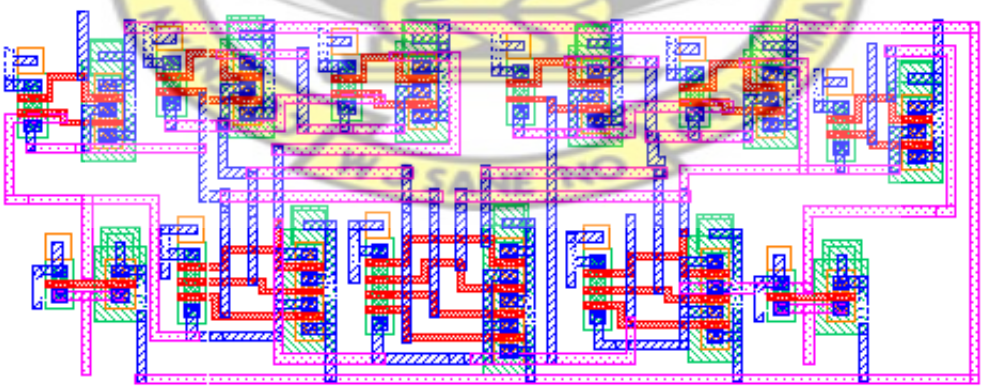


Figure A.2 - Phase/frequency detector layout

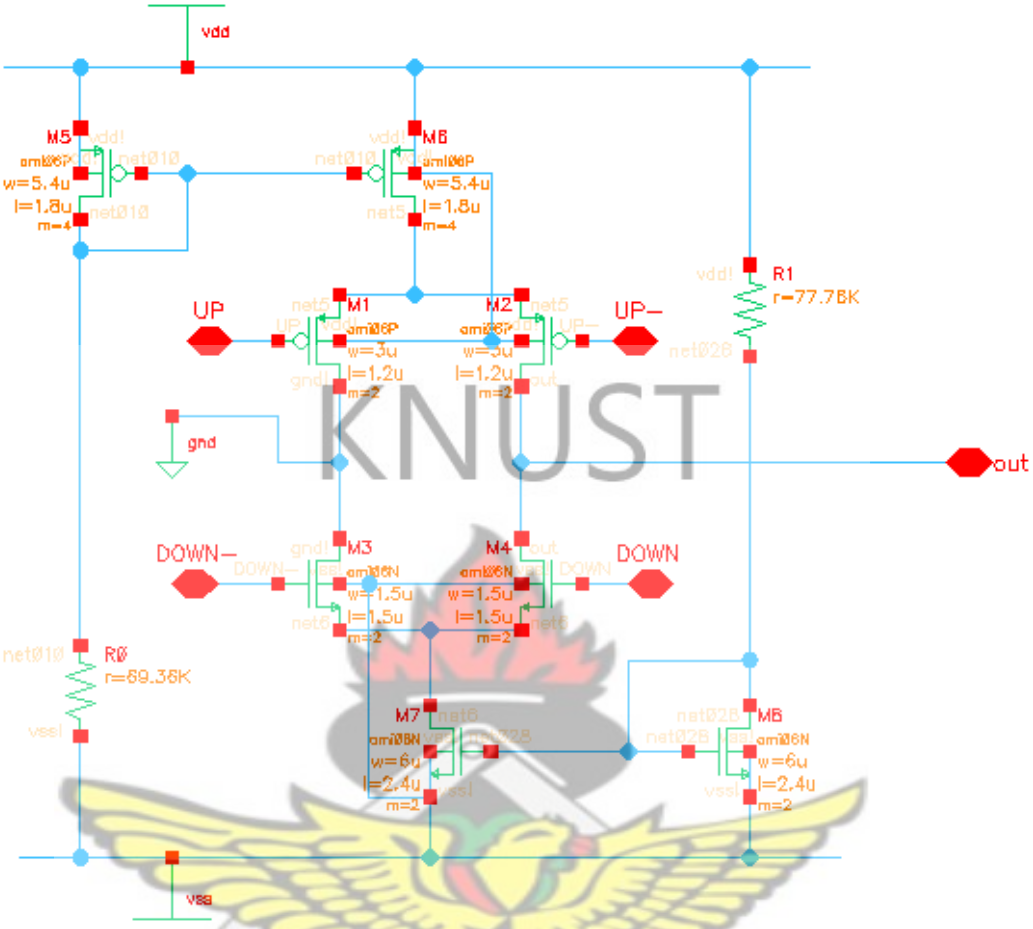


Figure A.3 - Charge pump schematic

Table A.1 - Charge pump layout floor plan

P3	P4	P1	P2
P4	P3	P2	P1
N1	N2	N3	N4
N2	N1	N4	N3

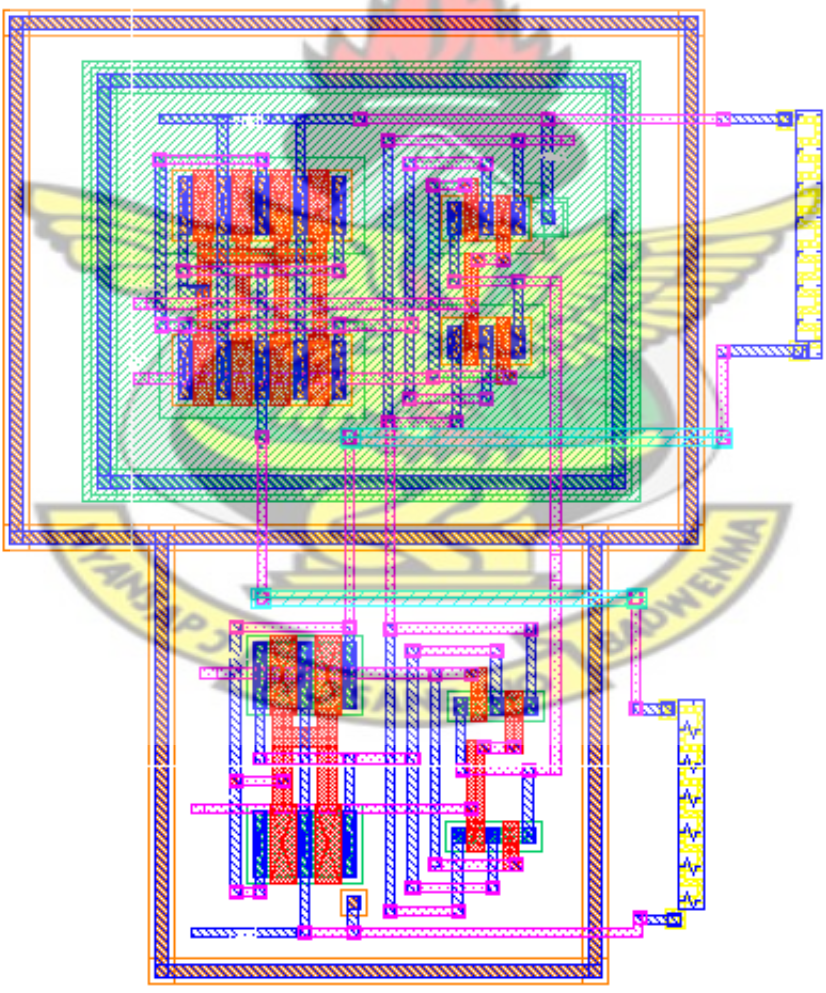


Figure A.4 - Charge pump layout

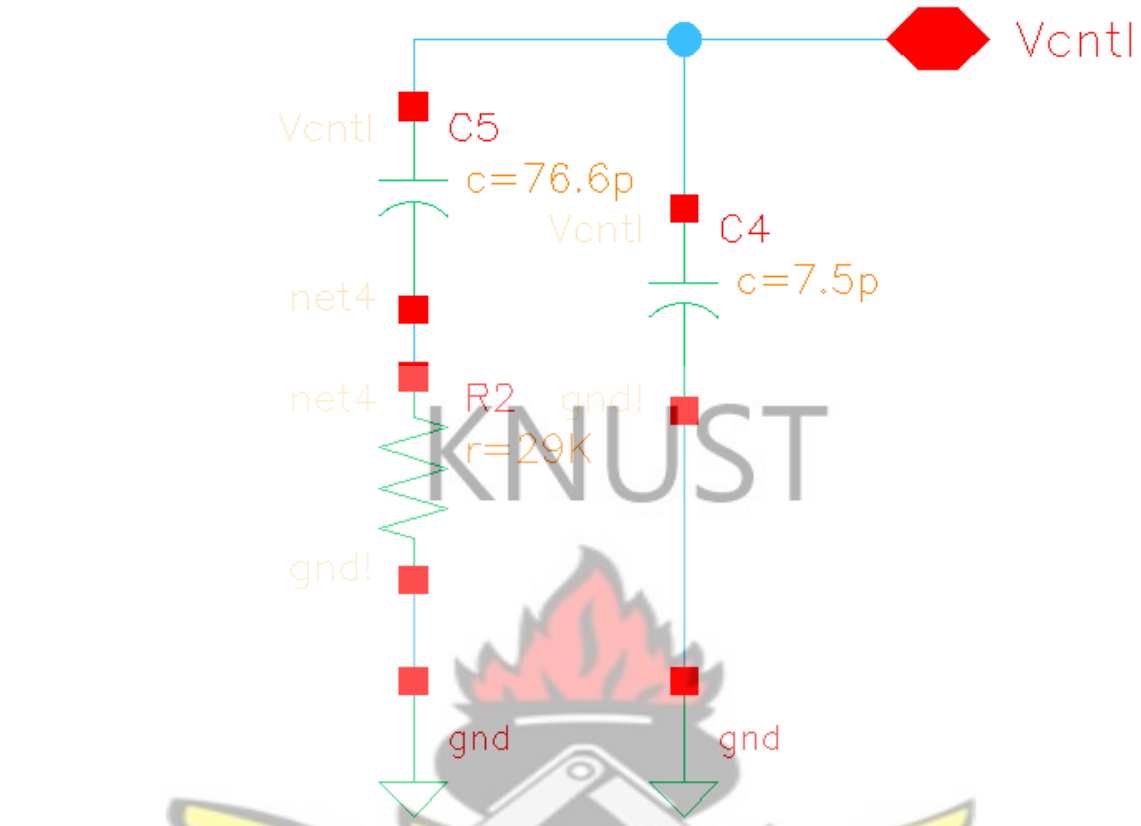


Figure A.5 - Filter schematic

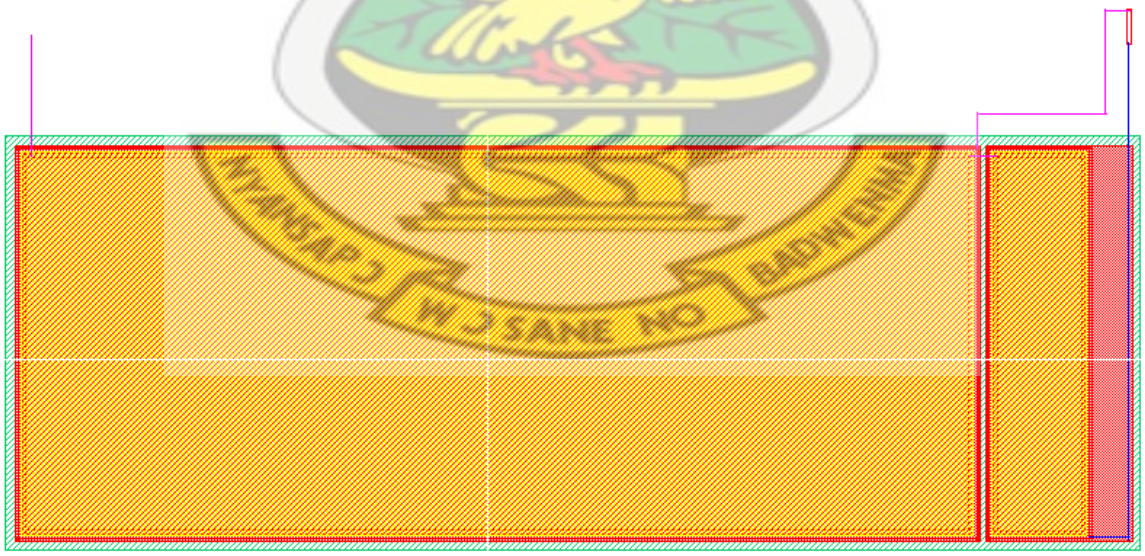


Figure A.6 - Filter layout

Table A.2 - VCO layout floor plan

D	P1	P2	P1	P1	P2	P1	P1	P2	P1	D
D	P2	P1	P1	P2	P1	P1	P2	P1	P1	D
D	P1	P2	P1	P1	P2	P1	P1	P2	P1	D
D	P2	P1	P1	P2	P1	P1	P2	P1	P1	D
D	P1	P2	P1	P1	P2	P1	P1	P2	P1	D
D	P2	P2	P1	P1	P2	P2	P1	P2	P2	D
D	P1	P2	P1	P1	P2	P1	P1	P2	P1	D
D	P2	P1	D	D	D	D	D	D	D	D
D	N1	N2	N1	N2	N1	N2	N1	N2	N1	D
D	N2	N1	N2	N1	N2	N1	N2	N1	N2	N1
D	N1	N1	N2	N1	N2	N1	N2	N1	N1	D

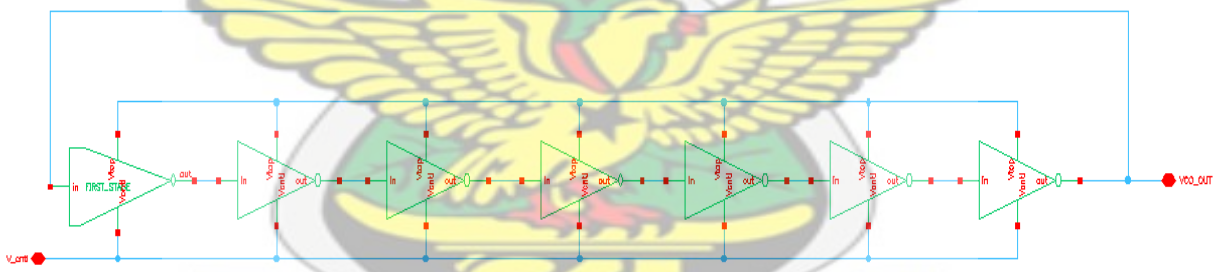


Figure A.7 - VCO Schematic

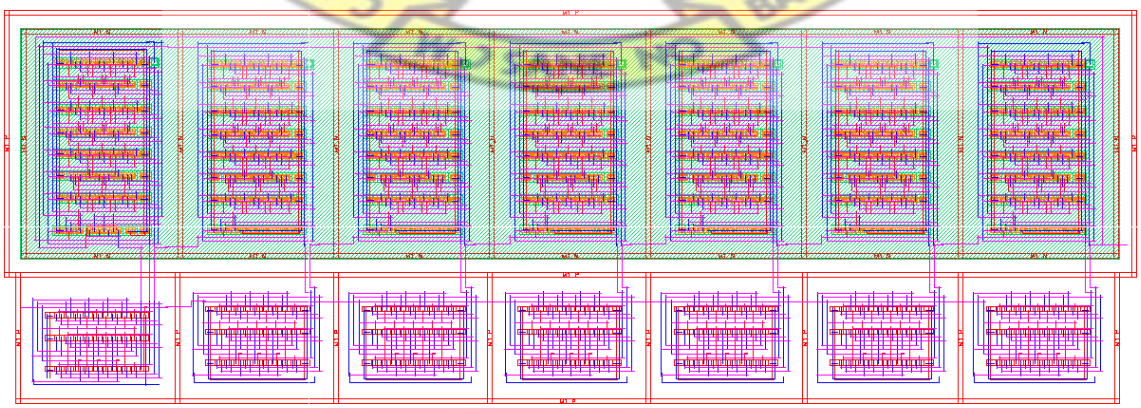


Figure A.8 - VCO layout

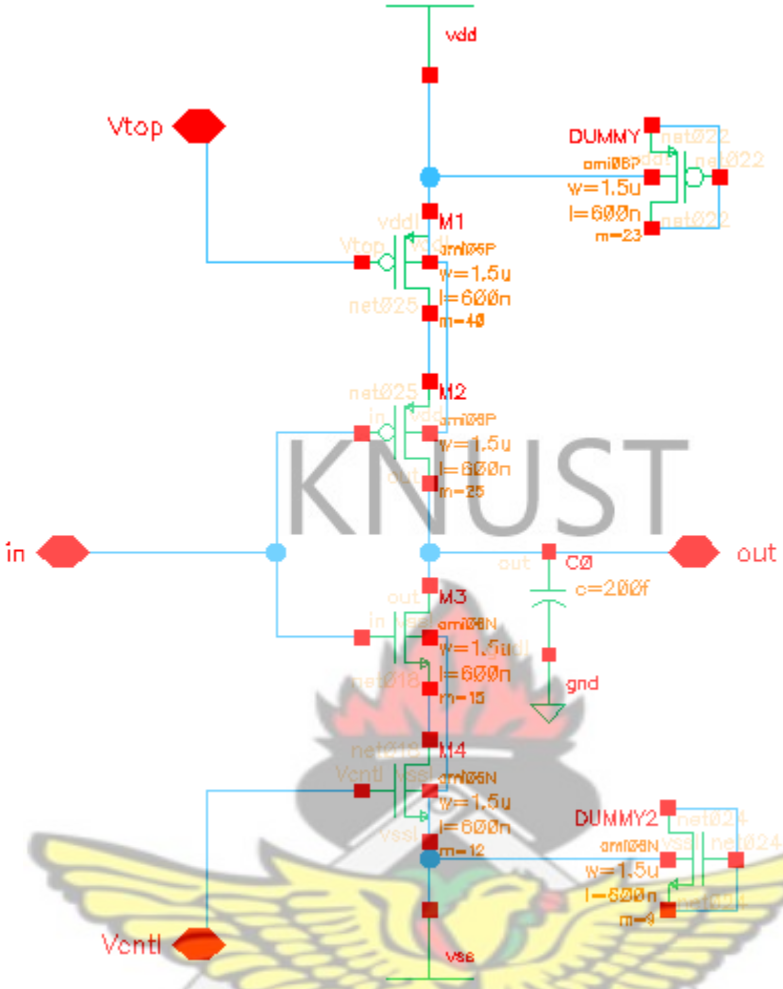


Figure A.9 - VCO inverting stage schematic

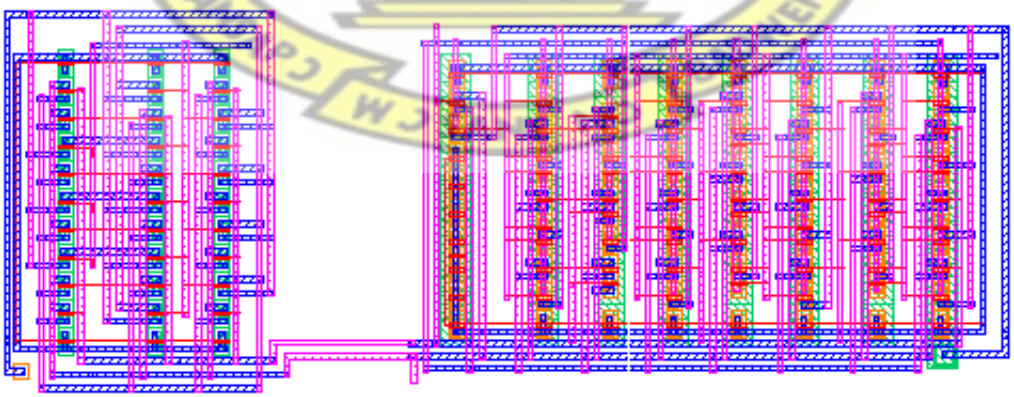


Figure A.10 - VCO inverting stage layout

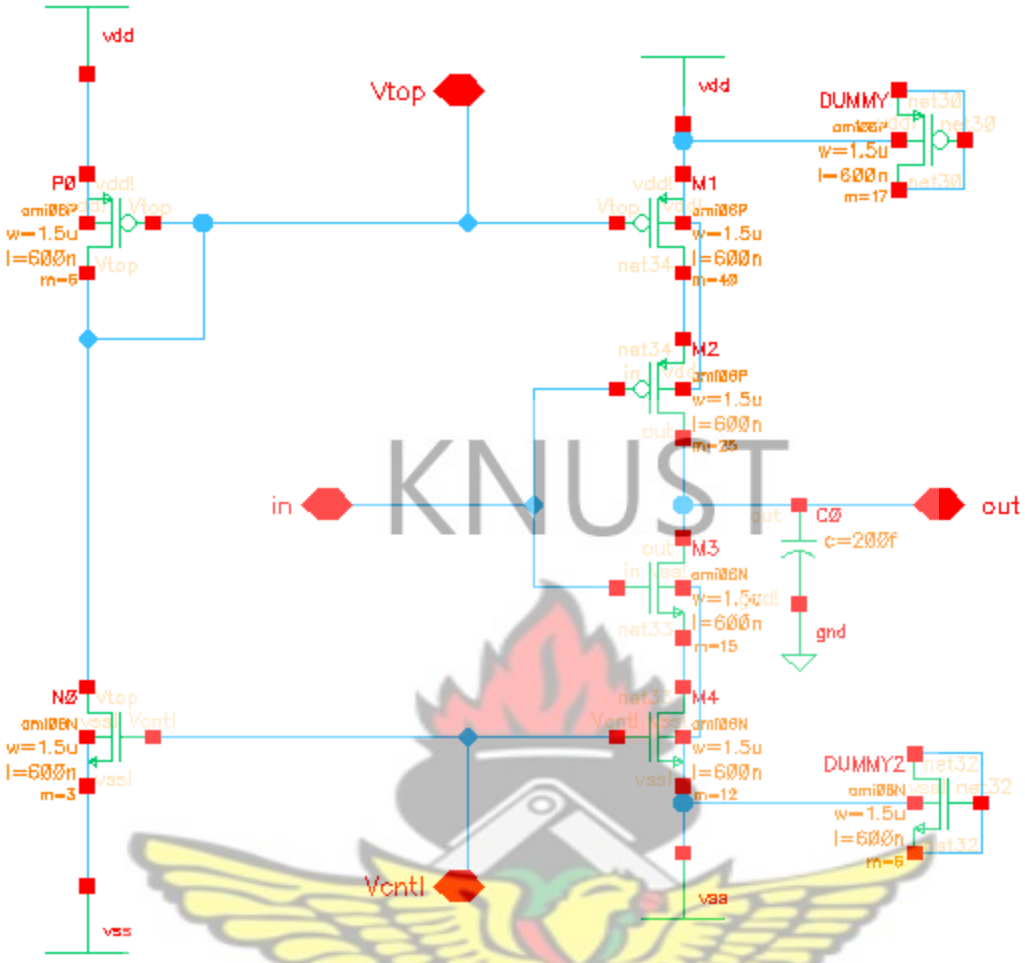


Figure A.11 - VCO inverting stage (first stage) schematic

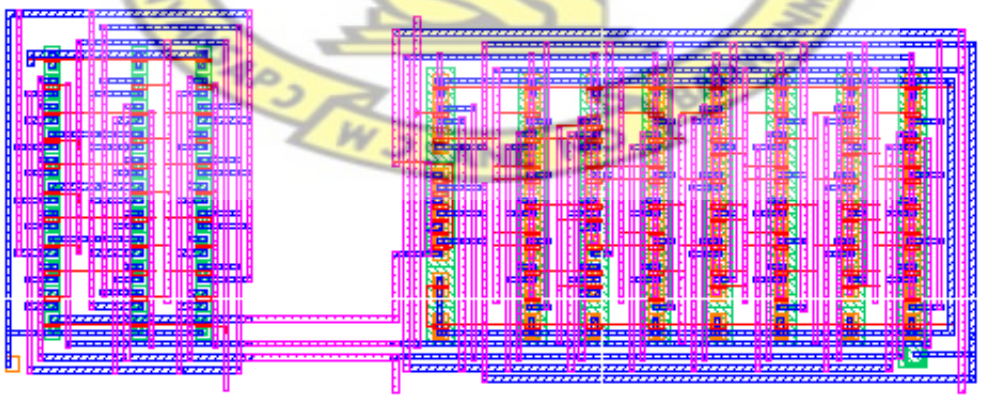


Figure A.12 - VCO inverting stage (first stage) layout

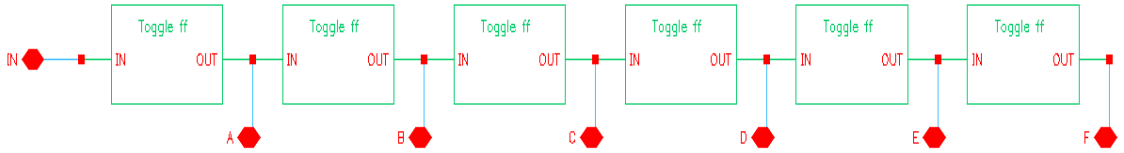


Figure A.13 – Divide-by-32 schematic

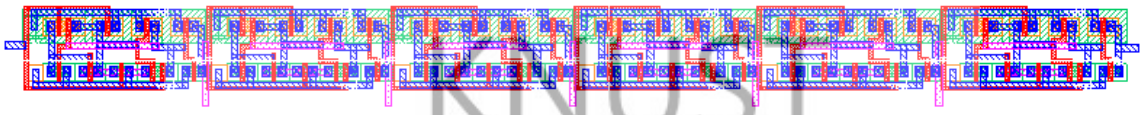


Figure A.14 – Divide-by-32 layout

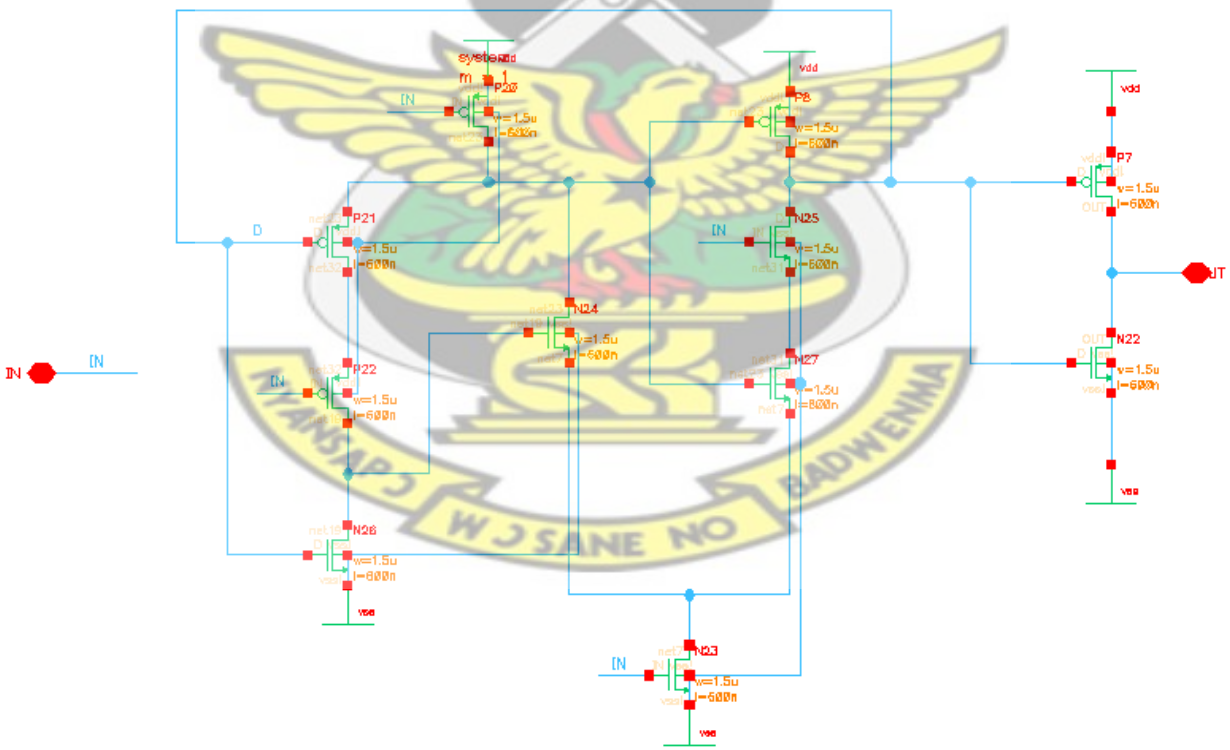


Figure A.15 - Divider flip-flop schematic

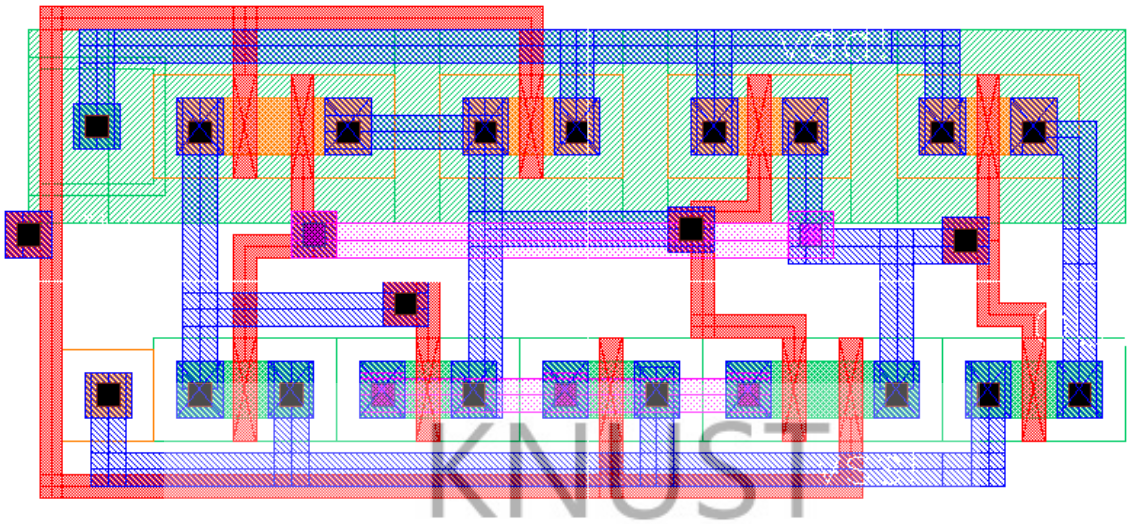


Figure A.16 - Divider flip-flop layout

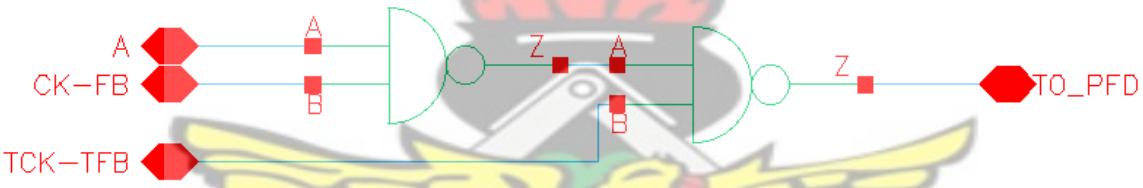


Figure A.17 - BIST/PLL switch schematic

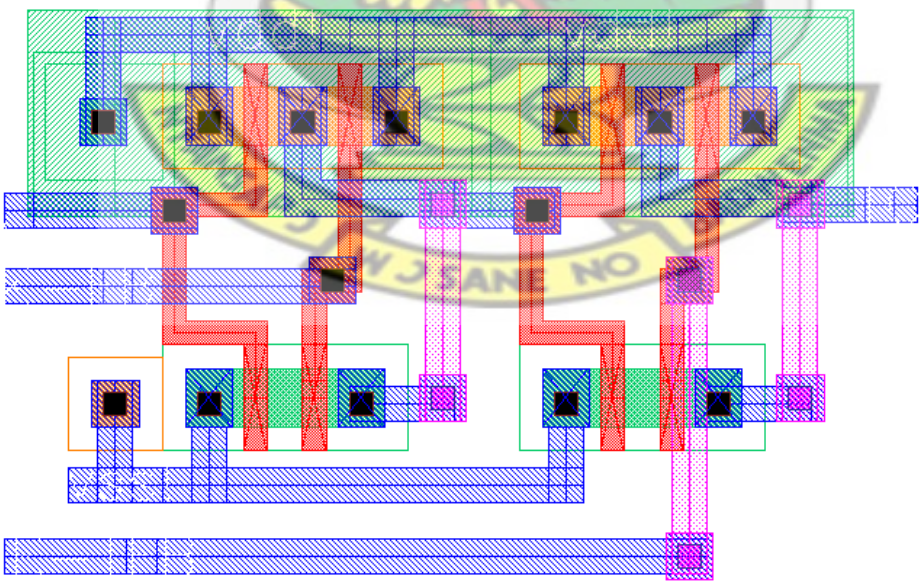


Figure A.18 - BIST/PLL switch layout

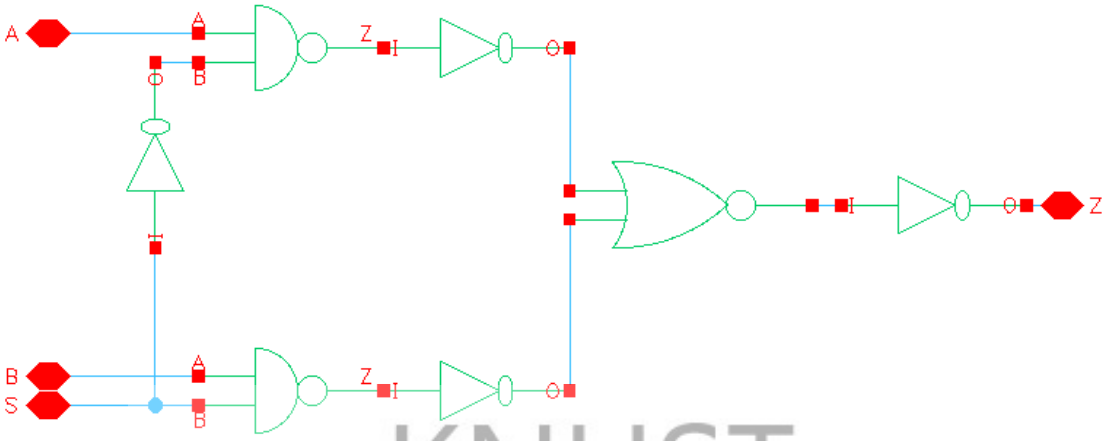


Figure A.19 - Multiplexer schematic

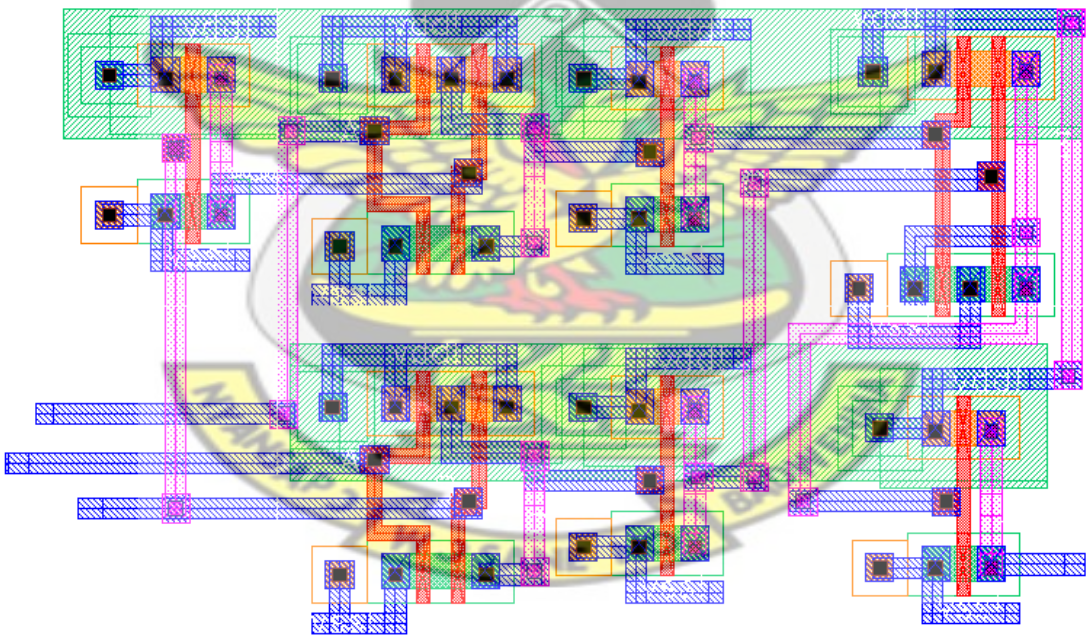


Figure A.20 - Multiplexer layout

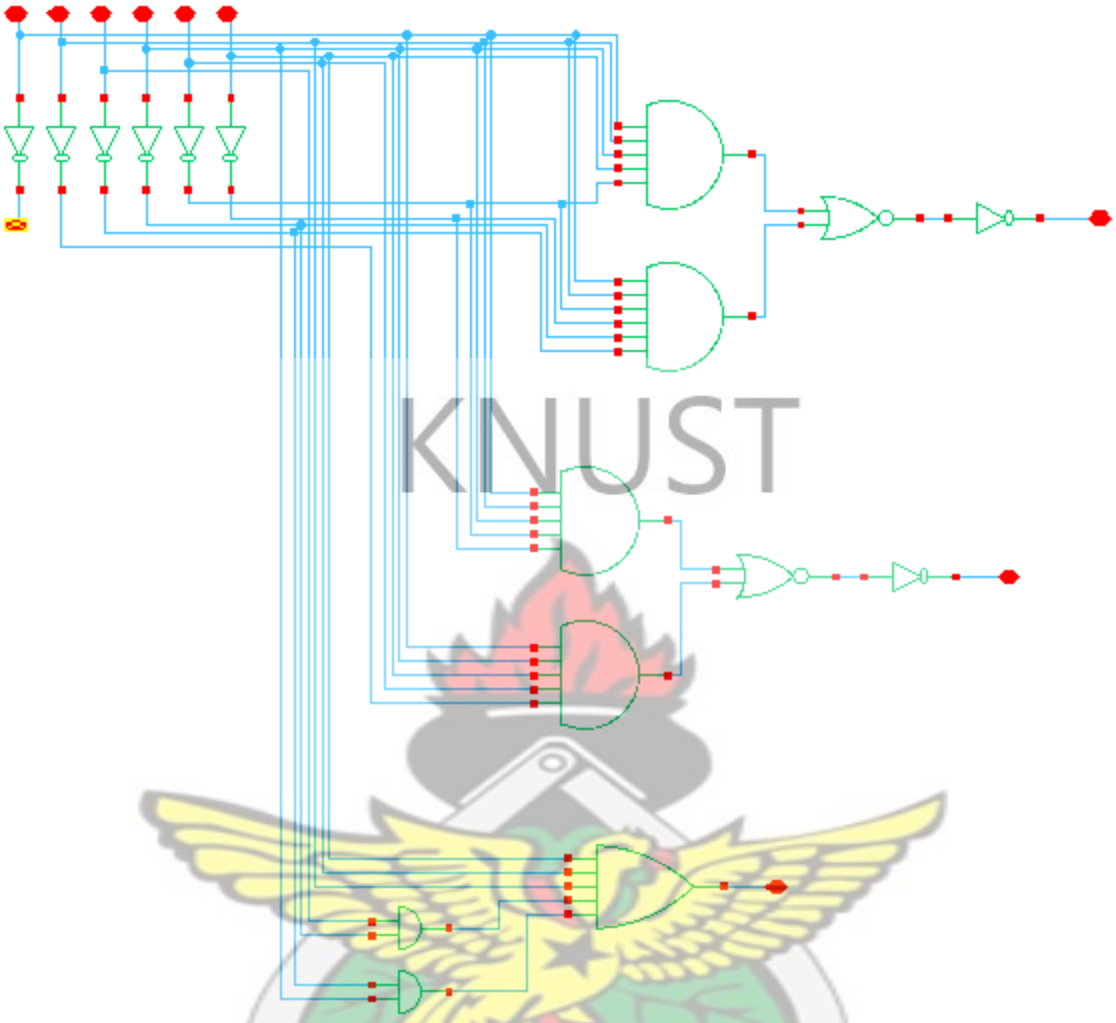


Figure A.21 - BIST input waveform generator schematic

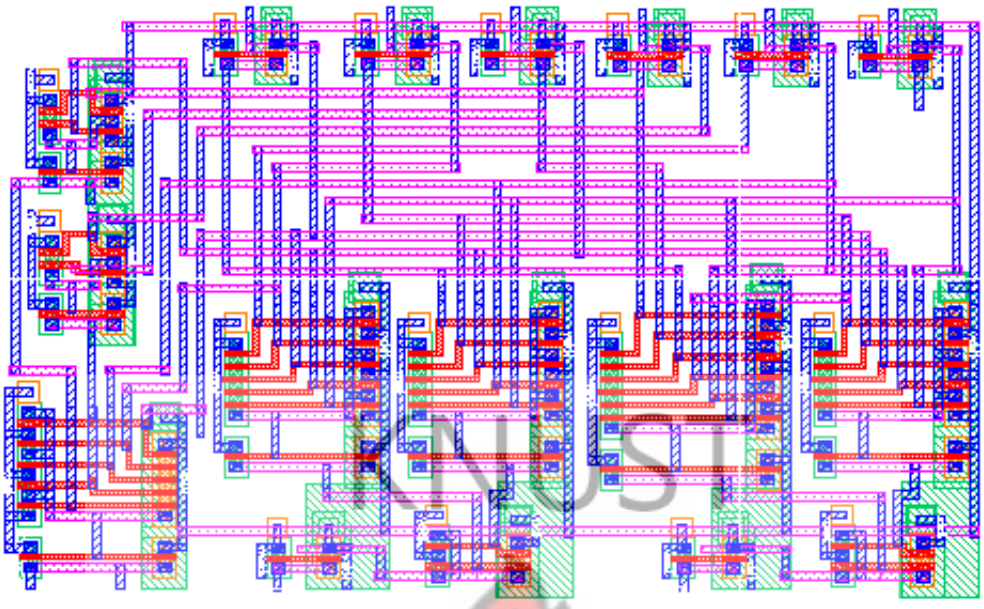


Figure A.22 - BIST input waveform generator layout

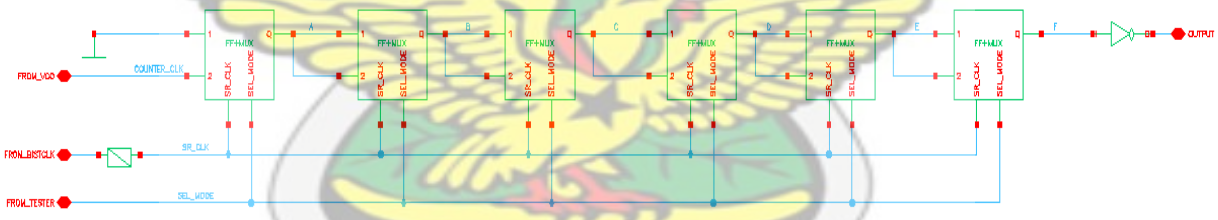


Figure A.23 - Response collector schematic

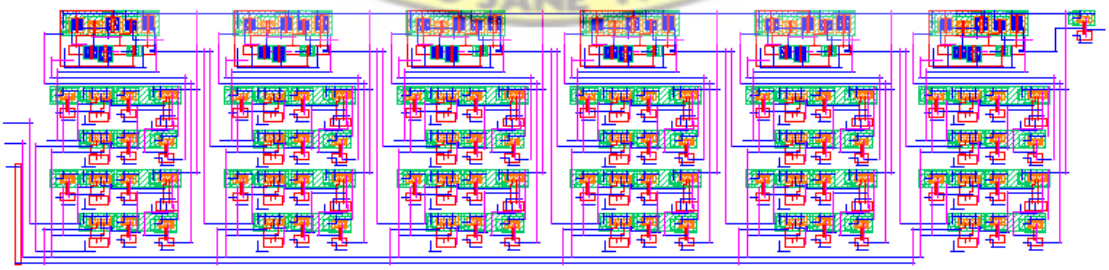


Figure A.24 - Response collector layout

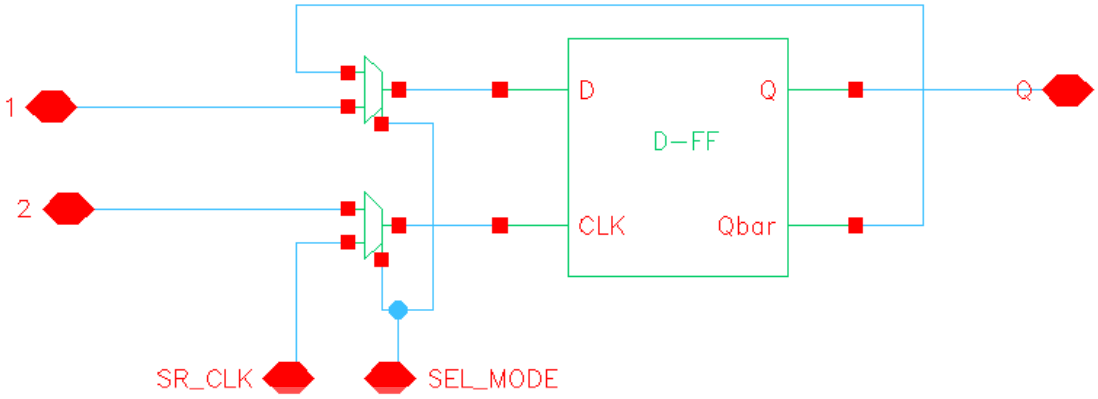


Figure A.25 - Response collector stage schematic

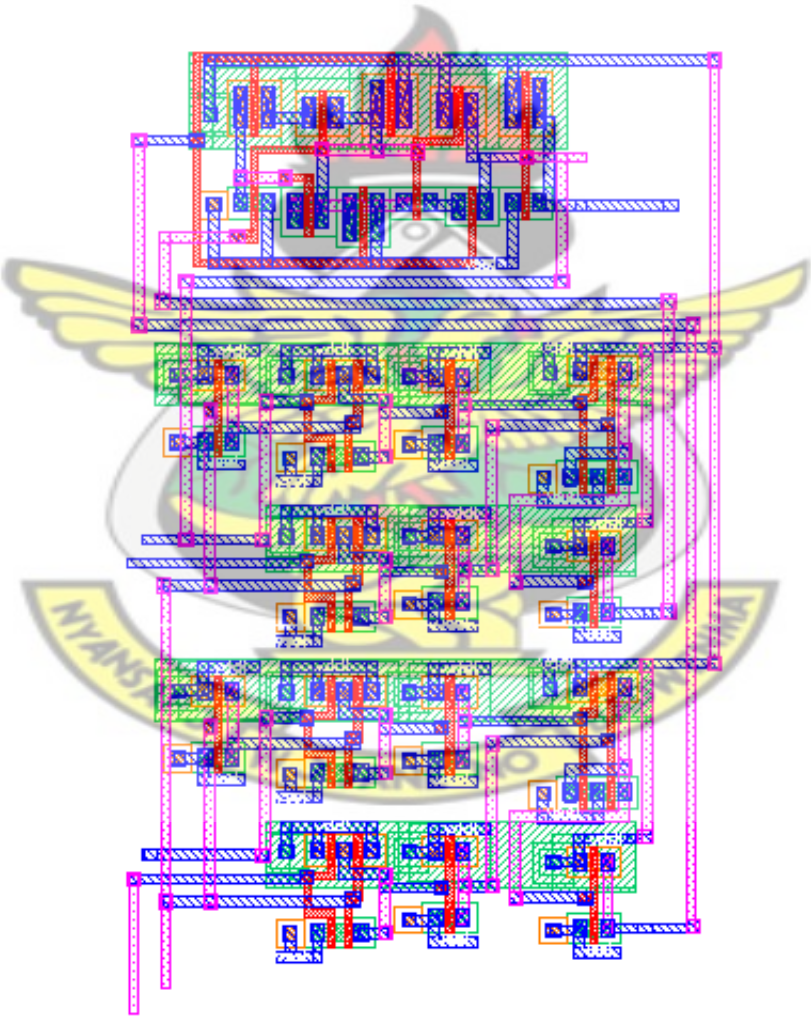


Figure A.26 - Response collector stage layout

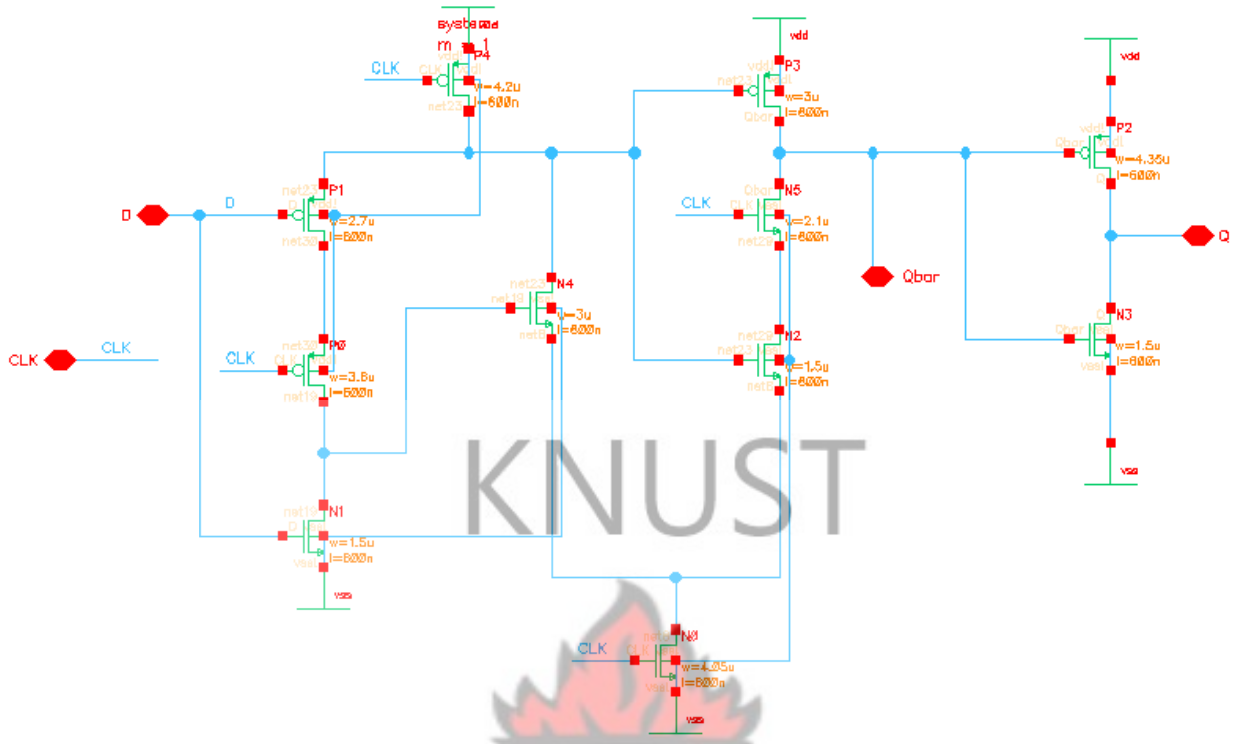


Figure A.27 - Schematic of flip-flop for Response Collector

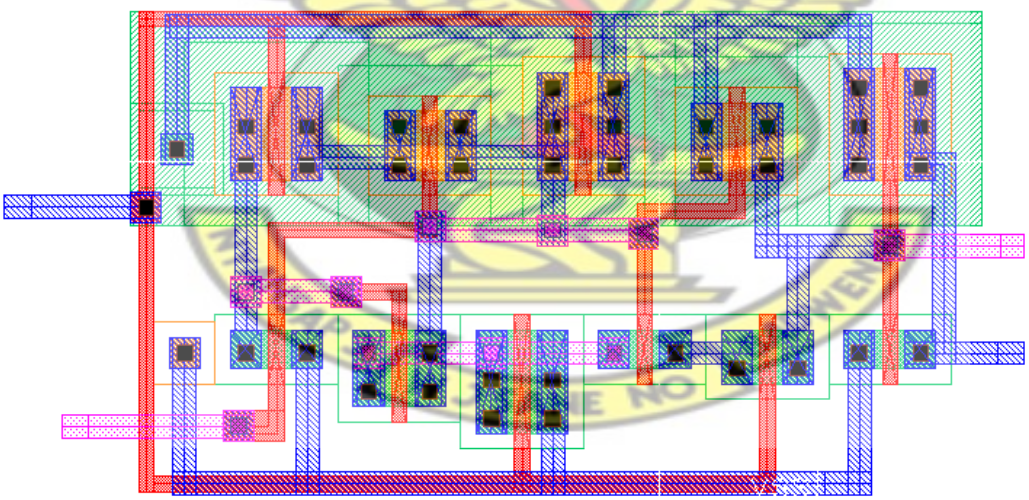


Figure A.28 - Layout of flip-flop for Response Collector

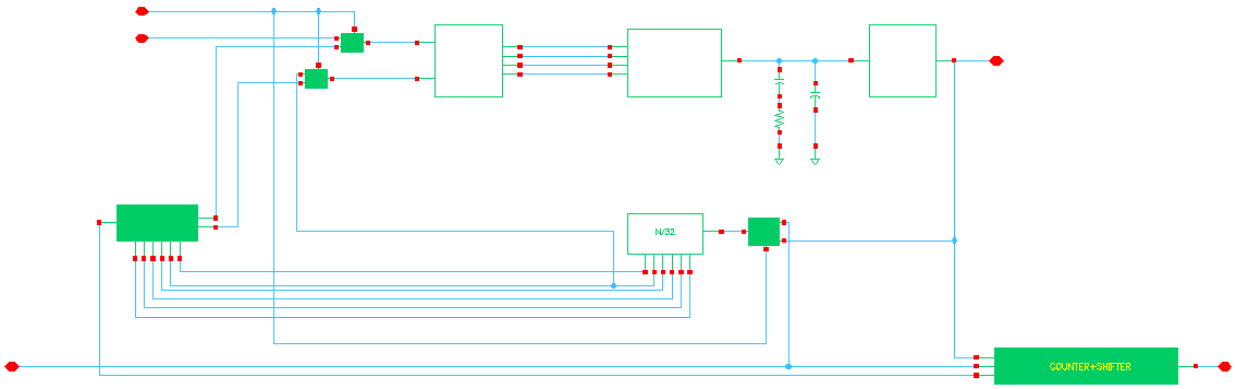


Figure A.29 - Schematic of PLL with BIST

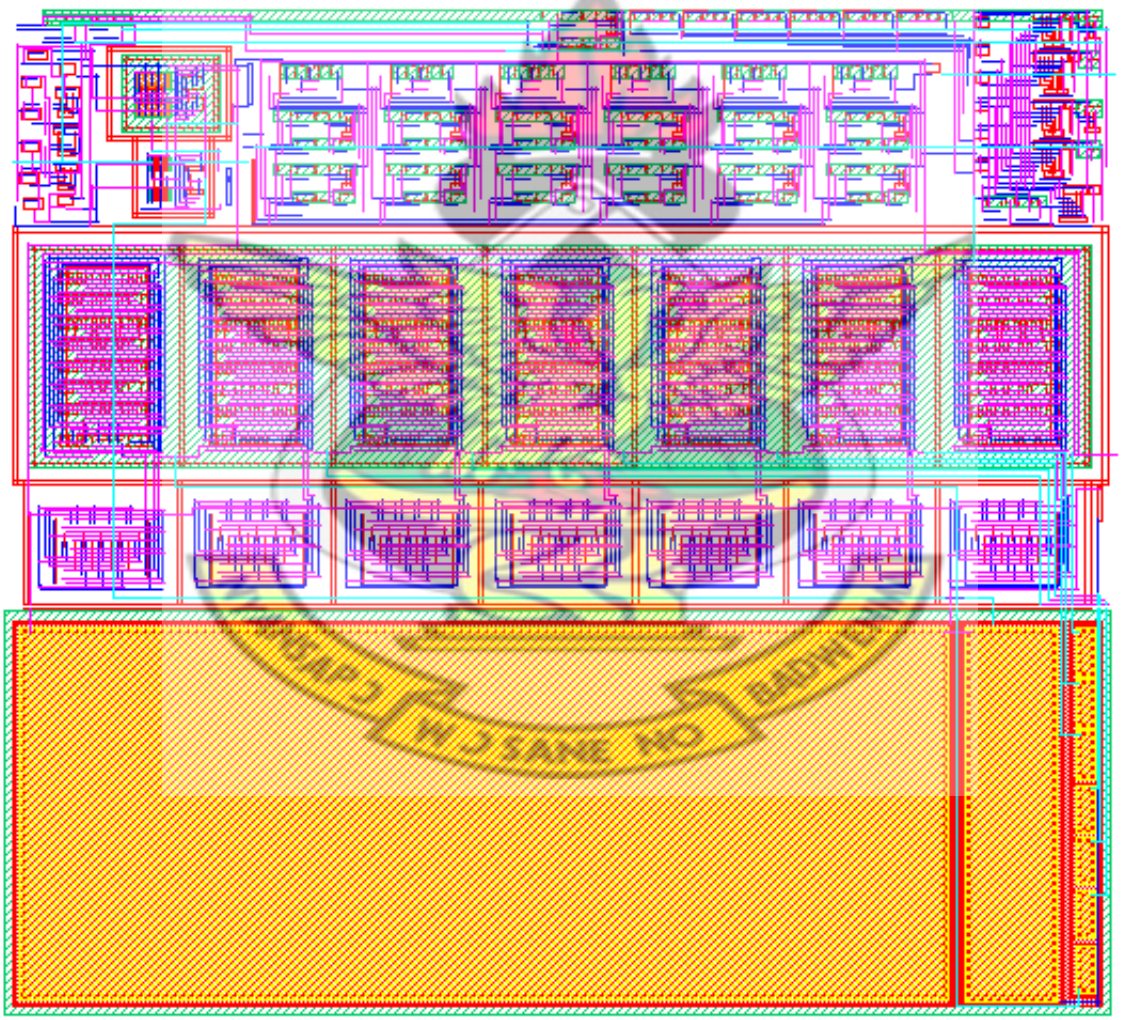


Figure A.30 - Layout of PLL with BIST

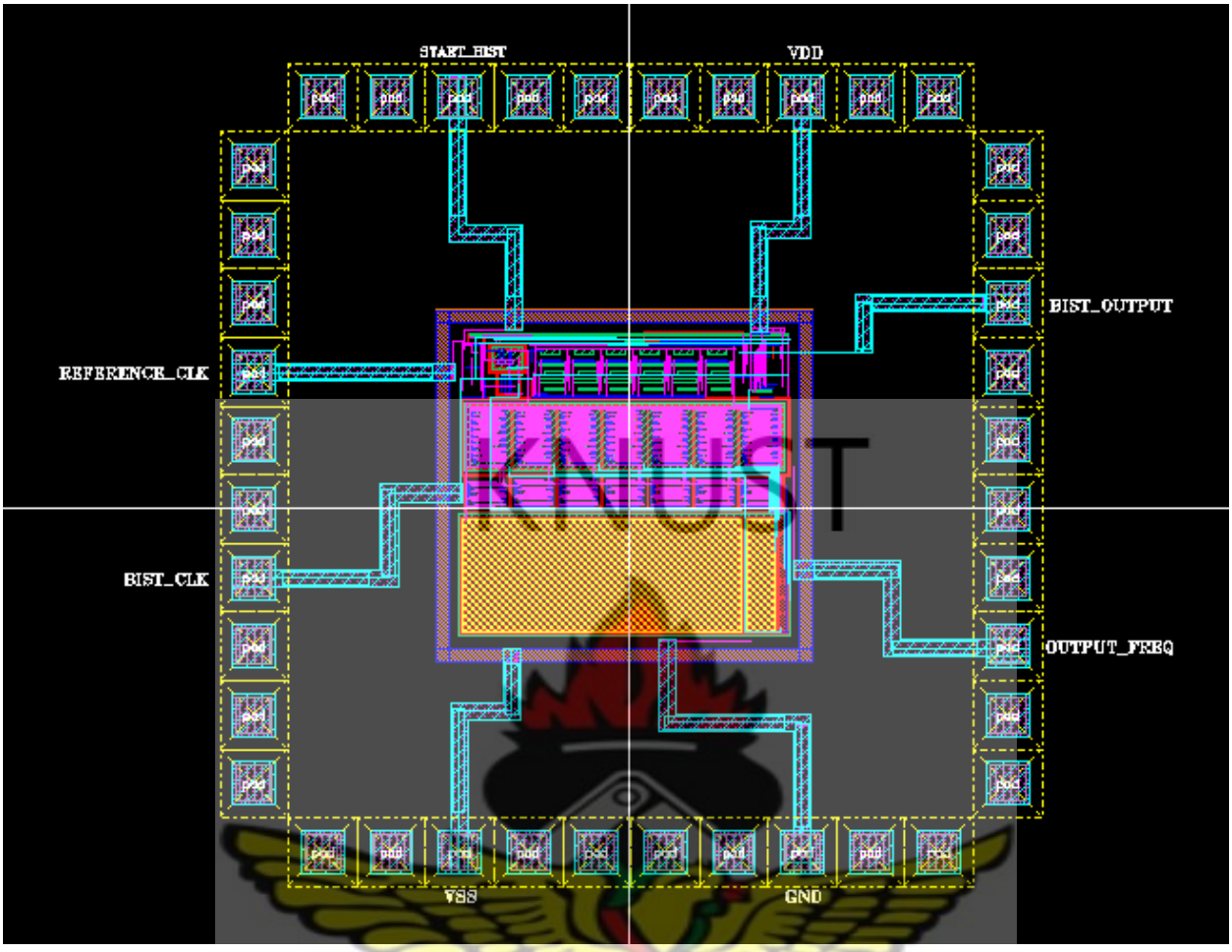


Figure A.31 - Final IC layout

Appendix B: BIST Input Waveform Generator Truth Table

Table B.1 - BIST input wave generator truth table

	A	B	C	D	E	F	TCK	TFB	SEL_MODE
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	1	0	0	1
2	0	0	0	0	1	0	0	0	1
3	0	0	0	0	1	1	0	0	1
4	0	0	0	1	0	0	0	0	1
5	0	0	0	1	0	1	0	0	1
6	0	0	0	1	1	0	0	0	1
7	0	0	0	1	1	1	0	0	1
8	0	0	1	0	0	0	0	0	1
9	0	0	1	0	0	1	0	0	1
10	0	0	1	0	1	0	0	0	1
11	0	0	1	0	1	1	0	0	1
12	0	0	1	1	0	0	0	0	0
13	0	0	1	1	0	1	0	0	1
14	0	0	1	1	1	0	0	0	1
15	0	0	1	1	1	1	0	0	1

16	0	1	0	0	0	0	0	0	1
17	0	1	0	0	0	1	0	0	1
18	0	1	0	0	1	0	0	0	1
19	0	1	0	0	1	1	0	0	1
20	0	1	0	1	0	0	0	0	1
21	0	1	0	1	0	1	0	0	1
22	0	1	0	1	1	0	0	0	1
23	0	1	0	1	1	1	0	0	1
24	0	1	1	0	0	0	0	0	1
25	0	1	1	0	0	1	0	0	1
26	0	1	1	0	1	0	0	0	1
7	0	1	1	0	1	1	0	0	1
28	0	1	1	1	0	0	0	0	1
29	0	1	1	1	0	1	0	0	1
30	0	1	1	1	1	0	0	0	1
31	0	1	1	1	1	1	0	0	1
32	1	0	0	0	0	0	0	0	0
33	1	0	0	0	0	1	0	0	1
34	1	0	0	0	1	0	0	0	1
35	1	0	0	0	1	1	0	0	1
36	1	0	0	1	0	0	0	0	1
37	1	0	0	1	0	1	0	0	1

38	1	0	0	1	1	0	0	0	1
39	1	0	0	1	1	1	0	1	1
40	1	0	1	0	0	0	0	0	1
41	1	0	1	0	0	1	0	0	1
42	1	0	1	0	1	0	0	0	1
43	1	0	1	0	1	1	0	0	1
44	1	0	1	1	0	0	0	0	0
45	1	0	1	1	0	1	0	0	1
46	1	0	1	1	1	0	0	0	1
47	1	0	1	1	1	1	0	1	1
48	1	1	0	0	0	0	1	0	1
49	1	1	0	0	0	1	0	0	1
50	1	1	0	0	1	0	0	0	1
51	1	1	0	0	1	1	0	0	1
52	1	1	0	1	0	0	0	1	1
53	1	1	0	1	0	1	1	0	1
54	1	1	0	1	1	0	0	0	1
55	1	1	0	1	1	1	0	0	1
56	1	1	1	0	0	0	0	0	1
57	1	1	1	0	0	1	0	0	1
58	1	1	1	0	1	0	0	0	1
59	1	1	1	0	1	1	0	0	1

60	1	1	1	1	0	0	0	1	1
61	1	1	1	1	0	1	1	0	1
62	1	1	1	1	1	0	0	0	1
63	1	1	1	1	1	1	0	0	1

References

1. GARG, K.; VERMA, V. S. Design Of Low Power Phase Locked Loop In Submicron Technology. **International Journal of Advanced Technology & Engineering Research**, v. 2, n. 2, March 2012. ISSN 2250-3536.
2. QUEMADA, C.; BISTUE, G.; ADIN, I. **Design Methodology for RF CMOS Phase Locked Loops**. London: Artech House, 2009.
3. HSIEH, G.; HUNG, J. C. Phase Locked Loop Techniques - A Survey. **IEEE Transactions On Industrial Electronics**, v. 43, n. 6, December 1996.
4. BEST, R. E. **Phase Locked Loops:** Design, Simulation and Applications. 5th. ed. New York: McGraw-Hill, 2003.
5. RAZAVI, B. **Monolythic Phase Locked Loops and Clock Recovery Circuits:** Theory and Design. New York: Wiley-IEEE Press, 1996.
6. NASH, G. Phase-Locked Loop Design Fundamentals. **Freescale Semiconductor**, n. AN535, February 2006.
7. KIM, S.; SOMA, M.; RISBUD, D. **An effective defect-oriented BIST architecture for high-speed phase-locked loops**. VLSI Test Symposium, 2000. Proceedings. 18th IEEE. New York: IEEE Press. 2000. p. 231-236.
8. HSU, C.; LAI, Y.; WANG, S. Built-In Self-Test for Phase-Locked Loops. **IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT**, v. 54, n. 3, p. 996-1002, June 2005.
9. SLIECH, K.; MARGALA, M. **A Digital BIST for Phase-Locked Loops**. Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS '08. IEEE International Symposium on. New York: IEEE. 2008. p. 134-142.
10. DASNURKAR, S.; ABRAHAM, J. **Closed-Loop Built-in-Self-Test for PLL Production Testing with Minimal Tester Resources**. Mixed-Signals, Sensors, and Systems Test Workshop, 2009. IMS3TW '09. IEEE 15th International. New York: IEEE. 2009. p. 1-5.
11. BOATENG, K. O. A New Defect-Oriented Method for Testing the PLL. **Journal of Engineering and Technology**, v. 1, p. 37-42, 2007.
12. STROUD, C. E. **A Designer's Guide to Built-In Self-Test**. Dordrecht: Kluwer Academic

Publishers, 2002.

13. STENSBY, J. L. **Phase-Locked Loops:** Theory and Applications. New York: CRC Press, 1997.
14. RAZAVI, B. **Phase-Locking in High-Performance Systems:** From Devices to Architectures. New York: Wiley-IEEE Press , 2003.
15. GARDNER, F. M. **Phaselock Techniques**. 2nd. ed. New York: John Wiley & Sons, 1979.
16. MARGARIS, N. I. **Theory Of The Non-Linear Analog Phase-Locked Loop**. Berlin: Springer-Verlag, 2004.
17. PALERMO, M. S. **A Multi-Band Phase-Locked Loop Frequency Synthesizer**. Texas A&M University. Texas. 1999.
18. FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAENI, A. **Feedback Control of Dynamic Systems**. 6th. ed. New York: Addison-Wesley, 1994.
19. STEPHENS, D. R. **Phase-locked Loops for Wireless Communications:** Digital, Analog and Optical Implementations. 2nd. ed. Boston: Kluwer Academic Publishers, 2002.
20. RAZAVI, B. **Design of Analog CMOS Integrated Circuits**. New York: McGraw Hill, 2001.
21. CHEEMA, H. M.; MAHMOUDI, R.; VAN ROERMUND, A. H. M. **60-GHz CMOS Phase-Locked Loops**. New York: Springer, 2010.
22. KROUPA, V. F. **Phase Lock Loops and Frequency Synthesis**. New York: John Wiley & Sons, 2003.
23. JOHNS, D. A.; MARTIN, K. **Analog Integrated Circuit Design**. Chicago: John Wiley & Sons, 1997.
24. TIEBOUT, M. **Low Power VCO Design in CMOS**. Berlin: Springer, 2006.
25. ENCINAS, J. B. **Phase Locked Loops**. New York: Chapman & Hall, 1993.
26. PARK, C. H.; KIM, B. A Low-Noise 900 MHz VCO in 0.6 µm CMOS. **IEEE Journal of Solid-State Circuits**, v. 34, n. 5, p. 586-591, 1999.
27. SIDIROPOULOS, S. et al. **Adaptive Bandwidth DLLs and PLLs using Regulated Supply CMOS Buffers**. Symposium on VLSI Circuits: Digest of Technical Papers. California: IEEE Press. 2000. p. 124-127.
28. SHI, X. **Design of Low Phase Noise Low Power CMOS Phase Locked Loops**. Neuchatel

University. Neuchatel. 2008.

29. PERROTT, M. H. **Techniques for High Data Rate Modulation and Low Power Operation of Fractional-N Frequency Synthesizers**. Massachusetts Institute of Technology. Massachusetts. 1997.
30. AGRAWAL, V. D.; KIME, C. R.; SALUJA, K. K. A tutorial on built-in self-test. I. Principles. **Design & Test of Computers, IEEE**, v. 10, n. 1, p. 73-82, 1993.
31. AGRAWAL, V. D.; KIME, C. R.; SALUJA, K. K. A tutorial on built-in self-test. 2. Applications. **Design & Test of Computers, IEEE**, v. 10, n. 2, p. 69-77, 1993.
32. RAJSKI, J.; TYSZER, J. **Arithmetic Built-in Self-Test for Embedded Systems**. New Jersey: Prentice Hall, Inc, 1998.
33. ABRAMOVICI, M. **Digital systems testing and testable design**. New Jersey: IEEE Press, 1990.
34. BUSNELL, M. L. **Essentials of electronic testing for digital, memory, and mixed-signal VLSI circuits**. Boston: Kluwer Academic Publishers, 2000.
35. STROUD, C. E. **A Designer's Guide to Built-in Self-Test**. Dordrecht : Kluwer Academic Publishers, 2002.
36. JOVANOVIC, G.; STOJCEV, M.; STAMENKOVIC, Z. A CMOS Voltage Controlled Oscillator with Improved Frequency Stability. **Scientific Publications of the State University of Novi Pazar**, v. 2, n. I, p. 1-9, 2010.
37. XU, C. et al. A Fully Integrated CMOS Phase-Locked Loop With 30MHz to 2GHz Locking Range and ±35 ps Jitter. **Analog Integrated Circuits and Signal Processing**, v. 36, n. 2, p. 91-97, 2003.
38. YOUNG, I. A.; GREASON, J. K.; WONG, K. L. A PLL Clock Generator with 5 to 110 MHz of Lock Range for Microprocessors. **IEEE Journal of Solid State Circuits**, v. 27, n. 11, November 1992.
39. RHEE, W. **Design of High-performance CMOS Charge Pumps in Phase-locked Loops**. Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on. New York: IEEE Press. 1999. p. 545-548.
40. YANG, S.; YOU, Y.; CHO, K. A New Dynamic D-flip-flop Aiming at Glitch and Charge Sharing

Free. **IEICE Transactions on Electronics**, v. E86-C, n. 3, p. 496-505, 2003.

41. GOLDMAN, S. **Phase-Locked Loop Engineering Handbook for Integrated Circuits**. London: Artech House, 2007.