Addis Ababa University

Addis Ababa Institute of Technology (AAiT)

School of Electrical and Computer Engineering

FPGA Based Optimum-Settling Automatic Gain Control Circuit Design for Multistage Amplification

BY

ABREHA TEKLU ASGEDOM

SUPERVISOR

Prof. MOHAMMED ABDO

A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment of the requirement for the Degree of Master of Science in Microelectronics Engineering

MARCH, 2018

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FPGA Based Optimum-Settling Automatic Gain Control Circuit Design for Multistage Amplification

By

Abreha Teklu Asgedom

ADDIS ABABA INSTITUTE OF TECHNOLOGY
APPROVAL BY BOARD OF EXAMINERS

Dr. Murad Ridwan	_____	**29/03/2018**
Chairman, Dept. of Graduate Committee	**Signature**	**Date**
Prof. Mohammed Abdo	_____	**29/03/2018**
Advisor	**Signature**	**Date**
Dr. Getachew Alemu	_____	**29/03/2018**
Internal Examiner	**Signature**	**Date**
Dr. Fetene Mulugeta	_____	**29/03/2018**
External Examiner	**Signature**	**Date**

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been fully acknowledged.

Abreha Teklu Asgedom	_____	March 29, 2018
Name	Signature	Date

Place: Addis Ababa

March 29, 2018
Date of submission

This thesis has been submitted for examination with my approval as a university advisor.

Prof. Mohammed Abdo	_____	March 29, 2018
Name	Signature	Date

ACKNOWLEDGEMENTS

I am greatly indebted to my thesis supervisor Prof. Mohammed Abdo for providing me with definite direction, professional guidance, and constant encouragement in many ways from the beginning till the end of my thesis work. I would also like to thank Dr. Getachew Alemu, former chair of Electronics Engineering in school of Electrical and Computer Engineering, for his follow up, advice and moral support as well as efficient coordination during thesis progress reports.

I acknowledge the coordination and guidance rendered by Dr. Yalemzewd Negash, dean of school of Electrical and Computer Engineering, and Dr. Murad Ridwan, postgraduate degree program coordinator, during the thesis work period. I would also like to appreciate the staff of school of Electrical and Computer Engineering as a whole and Electronics chair in specific for the valuable comments and suggestions they forwarded during my seminar one and two progress report presentations.

Above all I praise God who is always with me.

ABSTRACT

Many electronic applications require their input signal to be of constant amplitude. Especially, in communication systems, the transmitted signal may be faded and the amplitude of the received signal may be below the desired level due to multi-path fading and different positions. This thesis work solves problems of inconsistent amplitude signal arrival at receiving sides of electronic systems. Automatic Gain Control (AGC) circuits are mainly used in such applications to adjust the gain of a Variable Gain Amplifier (VGA) and to provide relatively constant amplitude signal to the receivers.

This work makes use of multi-stage amplification circuits to increase the dynamic range of the AGC. Three amplification stages are designed which increased the gain from 21 dB in single stage to 63 dB in three stages. Due to their flexibility better than ASICs as well as high performance speed and better power efficiency than microcontrollers, FPGAs are preferred in modern high speed technologies. Consequently, this thesis work used FPGAs to generate control voltages to adjust the gain of VGAs in the multistage amplification circuit. The designed AGC employs both feedback and feedforward loops to optimize the settling time. Feedback loop is used in input stage and feedforward loop is used in intermediate and output stages. When the amplitude level of the input signal diverges from the desired level, the FPGA generates a control voltage that adjusts the gain of the VGA and the result is an output signal with consistent amplitude level. A reference input of 25 mVpeak is considered and the AGC circuit is tested with input signals of amplitude below and above the chosen reference value. For all tested inputs a constant amplitude signal of 87 mVpeak is achieved.

Key-words: AGC, FPGA, VGA, feedback, feedforward, multistage, amplification

CONTENTS

LIST OF FIGURES AND TABLES

FIGURES

TABLES

ACRONYMS

1. AGC……..Automatic Gain Control
2. VGA……. Variable Gain Amplifier
3. FPGA……Field Programmable Gate Array
4. CPLD……Complex Programmable Logic Devices
5. BJT……...Bipolar Junction Transistor
6. FET……. Field Effect Transistor
7. MOS……Metal Oxide Semiconductor
8. MOSFET……derived from MOS and FET
9. CCD……Charge Coupled Devices
10. CMOS…… Complementary MOS
11. WLAN……Wireless Local Area Network
12. ASIC……. Application Specific Integrated Circuits
13. ADC……. Analog to Digital Converter
14. DAC……. Digital to Analog Converter
15. FSM……Finite State Machine
16. ASM……Algorithmic State Machine
17. VHDL…….....**V**ery High Speed IC **H**ardware **D**escription **L**anguage

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

Automatic Gain Control (AGC) circuits are used to adjust the gain, thereby providing relatively constant output amplitude so that circuits following the AGC require less dynamic range. They are usually employed in systems where the amplitude of an incoming signal varies over a wide dynamic range. The application domain of such systems is growing very rapidly from sensor interfaces to charge-coupled devices (CCD)/CMOS imagers, wire or wireless communications, disk drive read channels, medical, and multimedia systems, to name a few. Especially in the wireless communications, the signal strength level will be different due to the multi-path fading and different positions. The AGC is therefore used to adjust the received signal to a rated strength level [1].

As discussed in [2], AGCs can be implemented using feedback loop or feedforward loop. The input signal V_{IN} is amplified by a variable gain amplifier (VGA), whose gain is controlled by a signal V_C. In order to adjust the gain of the VGA to its optimal output level V_{OUT}, the AGC generally, first detects the strength level of the signal using the peak detector; it then compares this level with a reference voltage V_{REF} and finally, it filters and generates the required control voltage. This function can be performed by detecting the signal at the output of the VGA, so the architecture is called "feedback" AGC (Figure 1.1a), or at the input, in which case it is identified as "feedforward" AGC (Figure 1.1b).

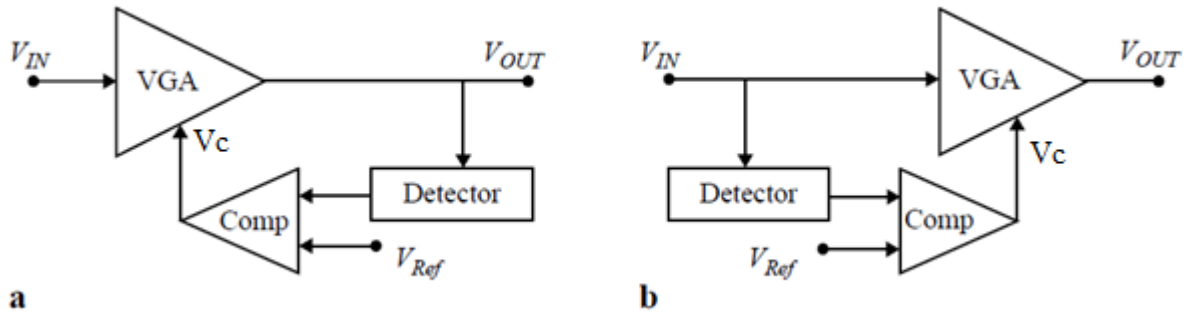


FIGURE 1.1: SIMPLIFIED BLOCK DIAGRAM OF FEEDBACK (A) AND FEEDFORWARD (B) AGCS

In addition to their linearity, feedback AGCs provide the level detector with less dynamic range but their settling time is slow, whereas feedforawrd AGCs have fast settling time. However, the level detector might be unmasked to high dynamic range of the input signal and the loop requires

higher linearity, since the feedback loop inherent linearity is now absent. Therefore, feedforward AGCs are difficult to realize due to high dynamic range requirement of the level detector circuit. Modern devices like WLAN receivers require fast-settling time as discussed in [3] and [4]; therefore, feedforward AGCs became preferable to conventional feedback AGCs in modern technologies which demand fast response.

In the past, AGC was being implemented fully in analog way as in Figure 1.2 below [5]. Later on, to obtain the advantages of digital over analog systems, engineers also implemented it digitally as presented in [6].

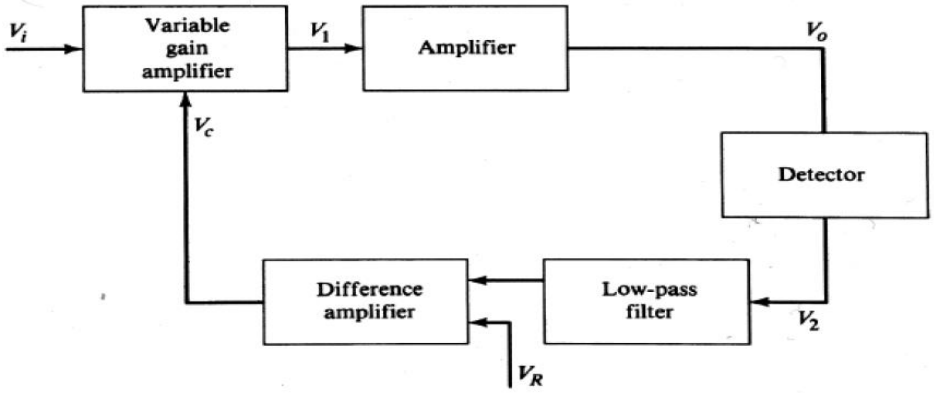


FIGURE 1.2: CONVENTIONAL ANALOG AGC BLOCK DIAGRAM

This thesis work used FPGAs to continuously generate control signals that adjust the gains of the amplifiers in order to tune the amplitude of the signal of interest to the desired (reference) level. FPGAs have better flexibility than ASICs as well as low power consumption and better performance speed than microprocessors [7].

1.2 PROBLEM STATEMENT

Many applications require the received signal to be of relatively constant amplitude. In disc drive reading systems, if they read varying amplitude signal, the person hearing may hear uneven sounds; in communication systems, the transmitted signal may be faded and the received signal will be below the desired level; in imaging devices, i.e. scanners, if the received signal from the scanning component is of inconsistent amplitude, the resulting image will be distorted, etc. This thesis work solves such inconsistencies in signal arrival at receiving sides of electronic systems by designing fast and flexible AGC.

1.3 OBJECTIVES

1.3.1 General Objective

To design an Automatic Gain Control (AGC) circuit for multistage amplification so that circuits following the AGC get constant amplitude signal.

1.3.2 Specific Objectives

- To design FPGA based AGC to make the AGC flexible, fast and power efficient
- To use both feedback and feedforward loop mechanisms to optimize the settling time
- To use multistage amplification to increase the dynamic range of the gain

1.4 SCOPE OF THE STUDY

Based on available resources and limited time frame, the scope of this thesis is limited to the following work.

- Design and cascade three basic VGA stages from MOSFET differential pair
- Develop a VHDL code that generates three control voltages for each VGA stage
- Construct embedded analog-digital AGC system from 3-stage VGA and an FPGA
- Software simulation and hardware implementation is done for the designed AGC circuit.

1.5 SIGNIFICANCE OF THE STUDY

The final outcome of this study is embedded FPGA based AGC circuit for three amplification stages. This AGC circuit is used to settle the variation in amplitude at input side of electronic systems and then provide them with constant amplitude signal. The design in this thesis work can be applied in areas which require arrival of constant amplitude signal at the receiver side; specially, in communication systems where the transmitted signal may be faded.

1.6 THESIS OUTLINE

This paper is organized in such a way that the first chapter introduces about AGCs and FPGAs, second chapter reviews the literature, third chapter discusses detailed design of the work, fourth chapter about materials used and the methodology applied for the design, fifth chapter discusses simulation and hardware implementation results, and the sixth chapter presents the recommendation for future works. The VHDL code that generates control voltages, Vc1, Vc2 and Vc3 is also placed in appendix.

CHAPTER 2: LITERATURE REVIEW

AGC was implemented in first radios for the reason of fading propagation (defined as slow variations in the amplitude of the received signals) which required continuing adjustments in the receiver's gain in order to maintain a relative constant output signal. Such situation led to the design of circuits, whose primary ideal function was to maintain a constant signal level at the output, regardless of the signal's variations at the input of the system [5]. This section assesses related and up-to-date works on design of AGCs. It is included as part of the thesis work to review previous and related works as well as to identify the gaps.

The idea behind an automatic gain control (AGC) circuit laid on design of variable gain amplifier (VGA) and control signal generating circuit. Many electrical and electronic engineers and researchers has focused on the efficient design of variable gain amplifiers (VGAs) which are basically applicable in design of AGC. Liu Hang, [8] designed a VGA in CMOS technology in partial fulfillment of his PhD. He proposed the idea of cell based approach to increase the dynamic range to desired level while designing VGAs.

The control signal generating circuit can be analog as in [1] or digital as in [6] and [9]. Implementing the gain adjust circuit in digital way provides the advantage of flexibility and computation accuracy. James R. Giebel, [6] under US patent number: 6,073,848 invented digital automatic gain control for multi-stage amplification circuits and he proposed implementation of the gain adjust circuit in digital state machine, i.e ASIC or microprocessor.

The gain adjust circuit can adjust the gain of the VGA either in feedback as in [1] and [6], or feedforward loops as in [3] and [4]. Feedback AGCs are linear and provide the peak detector with less dynamic range; however, they have relatively slow settling/response time; whereas, feedforward AGCs have fast settling time at the cost of providing the peak detector with high dynamic range [2].

AGC can be designed and applied for specific applications. They can be used in WLAN, LAN and Bluetooth receivers as elaborated in [1], [3], [9], [10], and [11]. They can also be used in software defined radios as discussed in [12] and [13]. AGC is also essential in Global Navigation Satellite System (GNSS) receivers, like GPS receivers due to the complicated application environment, such as indoor circumstance, temperature variation, and radio frequency interference (RFI) as discussed by Jin Gang et al [14].

X. Chen and Min Du, [15] designed an AGC for multi-channel electrochemical detection system which provides fast electrochemical data acquisition system by amplifying and adjusting the amplitude signal acquired from detection of the very low quantities, like DNA and RNA. As they discussed in their paper, electrochemical approaches offer an attractive detection modality and play a significant role in biomedical analysis, industrial quality control and environmental analysis. Erick Salas and Pablo Alvarado, [16] also designed an AGC in an application for environmental protection which detects audio signals from like chainsaws and guns, thereby preventing the forest by informing the concerned legal body.

Out of the aforementioned related works, the AGC design proposed by James R. Giebel, [6] is more close to this thesis work in many ways; however, the technology used to design the VGA, i.e MOS or BJT is not specified. This thesis work used MOS differential pair to design the VGA due to its capability to remove common mode noise signals and better coupling capability. Besides, due to separation of the gate from substrate by SiO2 acting as insulator, MOSFETs have less power dissipation than BJTs. In addition, this thesis work differs from the works stated above in such a way that both feedback and feedforward loops are employed to design the AGC to optimize the response/settling time. Moreover, the gain adjust circuit is digital and is implemented in FPGA for better flexibility than ASICs as well as less power consumption and better performance speed than microcontrollers/microprocessors.

CHAPTER 3: SYSTEM DESIGN AND DESCRIPTION

The general block diagram representation of the designed FPGA based AGC circuit for three amplification stages is shown in Figure 3.1 below. At the top is variable Gain Amplifier (VGA) whose gain can be varied by control signal Vc to deliver a signal with desired amplitude level. To reject noise and for better coupling in multistage amplification, a differential amplifier is used for design of the VGA. The output from each stage VGA is differential ac signal swinging on dc output signal. To remove the common mode dc voltage and measure only the differential ac output on each stage and to amplify it for readability of small signal values, an instrumentation amplifier is used. In order to generate a control signal that adjusts the gain of the VGA, it is necessary to detect the peak values of the signal amplitude at each stage. For this purpose, a peak detector circuit is designed and used which takes output of each instrumentation amplifier and outputs the peak values. The output at the peak detector circuit is a peak dc voltage which will be converted to digital equivalent using Analog to Digital Converter (ADC) circuit in the FPGA. Comparing the detected peak value with the desired reference value, the FPGA computes and generates a control signal that adjusts the gain of the related VGA. Since the control signal generated by the FPGA is digital, it will be converted to equivalent analog value by Digital to Analog Converter (DAC). Finally, the output of the DAC is fed to VGAs to adjust over all multistage gain.

Both feedback and feedforward topologies of closed loop are used. As discussed in [2], feedback loop has the advantage of better linearity and since the peak detector circuit senses at the output, it is not exposed to high dynamic range signals. However, since the amplitude level of the signal is being detected at the output after it passes the amplifier, and feeds it back to the input, it has a delay which results in slow settling time. Feedforward loop has the advantage of fast settling time because it senses the amplitude level at the input before it passes the amplifier and feeds it forward which minimizes delay effect of the amplifier on the settling time. However, it lacks linearity and exposes the level detector to high dynamic range. The first stage is highly probable and the remaining middle and output stages are less probable to be exposed to high dynamic range signal. To utilize the strong sides of both loops, feedback loop is used at input stage and feedforward loop at the middle and output stages which results in optimum settling time as well as realizable design.

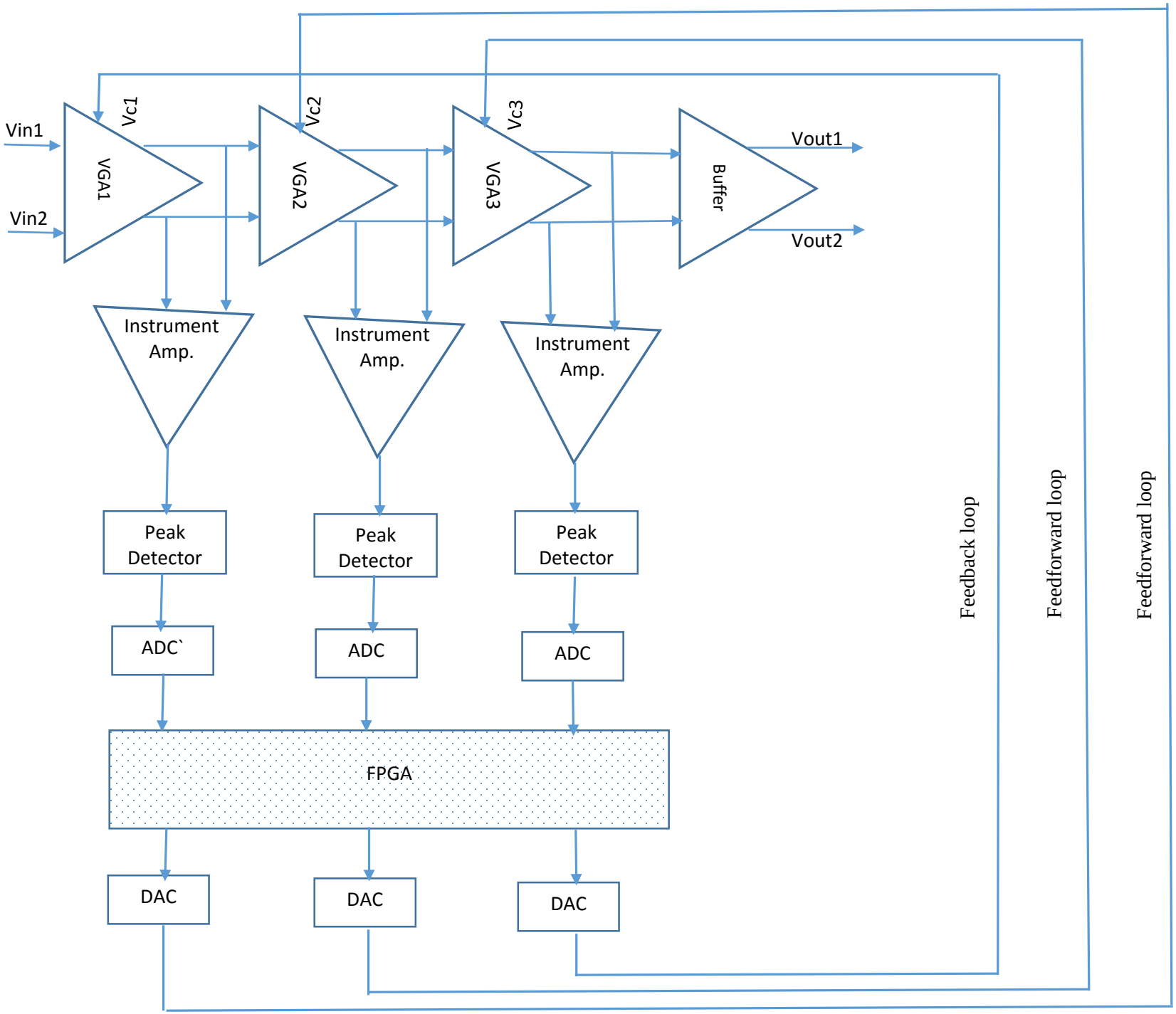


FIGURE 3.1: BLOCK DIAGRAM REPRESENTATION OF THE DESIGNED FPGA BASED AGC CIRCUIT

The process of gain adjustment using feedback and feedforward loops is as follows:

Stpe1: feedback loop

a) Initially, the circuit starts sensing at the last stage, if there is variation as compared to the reference value, it generates control signal Vc1 and feeds it back to first stage VGA.
b) If the gain cannot be adjusted by gain range of first stage VGA, the AGC tries to adjust the amplitude level up to the possible level it can, and continues to step2.

Step2: feedforward loop at stage two VGA

a. The AGC circuit checks if there is a variation from the desired amplitude level at output of first stage VGA (input of second stage VGA). If there is, it generates control signal Vc2 and feeds it forward to second stage VGA.
b. If the gain cannot be adjusted by gain range of second stage VGA, the AGC tries to adjust the amplitude level up to the possible level it can, and continues to step3.

Step3: feedforward loop at stage three VGA

a. The AGC circuit checks if there is a variation from the desired amplitude level at output of second stage VGA (input of third stage VGA). If there is, it generates control signal Vc3 and feeds it forward to third stage VGA.
b. If the gain cannot be adjusted by gain range of third stage VGA, the AGC tries to adjust the amplitude level up to the possible level it can, and loops to step1.

The whole computation of the required gain adjustment, and generation of control voltage for adjusting the gain is done in one clock cycle of the FPGA for each of the above steps. In the subsequent topics, basic building blocks and components that are used in the design are explored and discussed.

3.1 VARIABLE GAIN AMPLIFIER DESIGN

Variable Gain Amplifiers (VGAs) find application in systems where the signal amplitude may experience large variations and hence requires inverse changes in the gain. Since the objective of this MSc thesis research is to design an automatic gain control circuit to deliver constant amplitude signal to circuits that require constant amplitude signal, VGAs have vital role in the AGC circuit design. Applicable VGA can be designed using MOS differential pair as shown in Figure 3.2 below. [17].

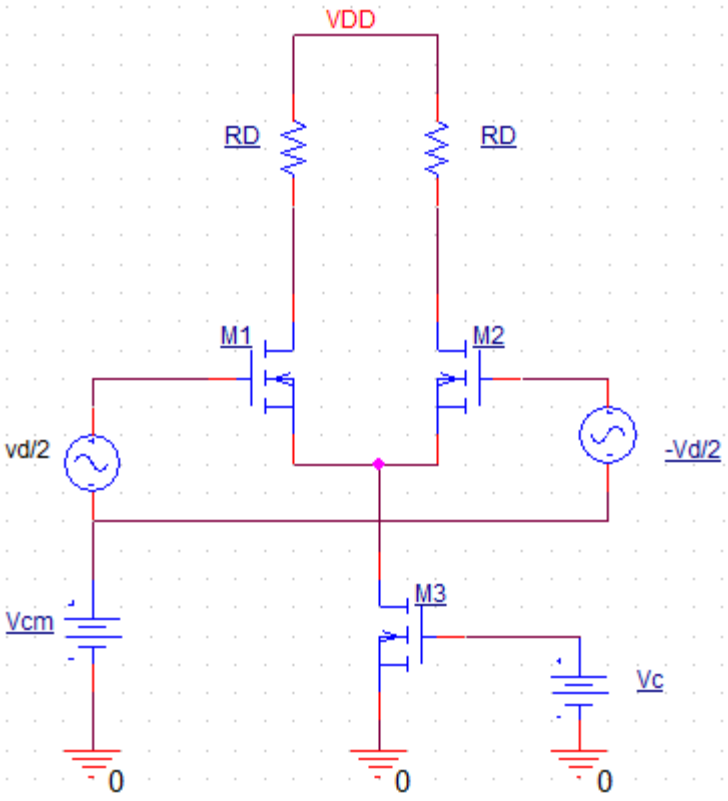


FIGURE 3.2: VARIABLE GAIN AMPLIFIER

MOSFETs M1 and M2 act as derivers of the basic differential pair while M3 acts as biasing current source whose current can be varied by the control gate voltage Vc. R_D is a load resistor whose first terminal is connected to drain of MOSFETs M1 or M2, and second terminal connected to biasing voltage V_{DD}. V_{cm} is a common mode dc biasing at the gates of M1 and M2.

Since the VGA is designed from MOS differential pair, no need of coupling capacitors for multistage amplification. The output of each differential pair in each stage is fed to the instrumentation amplifier which then removes the common mode dc signal and amplifies the differential ac signal. The OPAMPs in the instrumentation amplifier has limited input and output voltage range, for example uA741 OPAMP has ± 15V and ± 10V input and output voltage ranges respectively. Consequently, V_{cm} must not be too positive and too negative. If source of M3 is connected to ground, Vcm will be too positive and when it is amplified, it may go beyond input range of the OPAMP.

Since $V_{GS}>V_{TN}$, to decrease $|V_G|=|Vcm|$, negative bias $V_{SS}=-V_{DD}$ can be applied at the source of M3. By applying V_{DD} and V_{SS}, not only minimizes |Vcm| but also increases the output

voltage swing which means increasing the gain range. The modified VGA circuit based on the aforementioned discussion and analysis is shown in Figure 3.3 below.

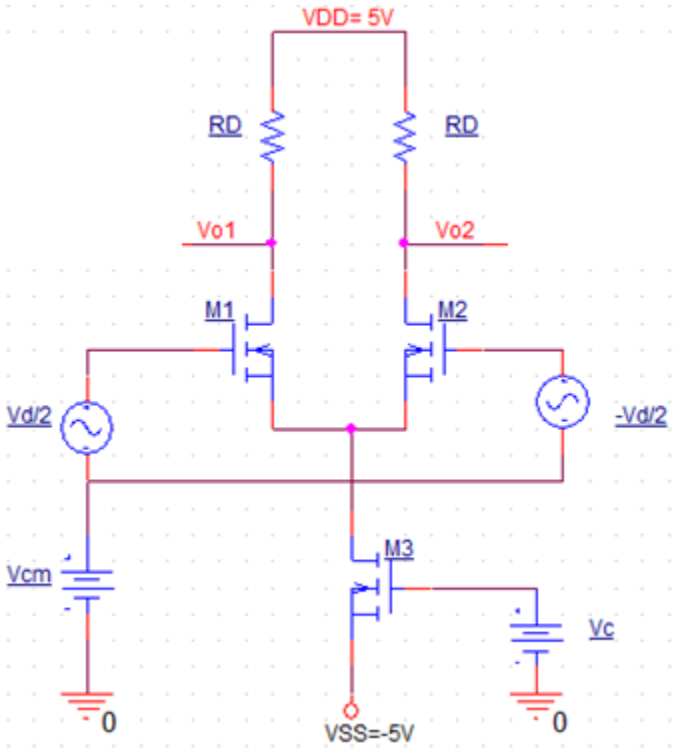


FIGURE 3.3: DESIGNED VGA WITH APPROPRIATE DC BIASING

For the above designed VGA, the following components and values are selected:

- The input frequency f= 1KHz
- V_{DD}=3V, and V_{SS}=-3V. V_{DD}=5V and V_{SS}=-5V is not selected as this makes the control voltage Vc<0 which creates complexity for unsigned arithmetic operations in the FPGA.
- IRF840 NMOS. Because it is available in the lab

Using the above components, the VGA will be as follows:

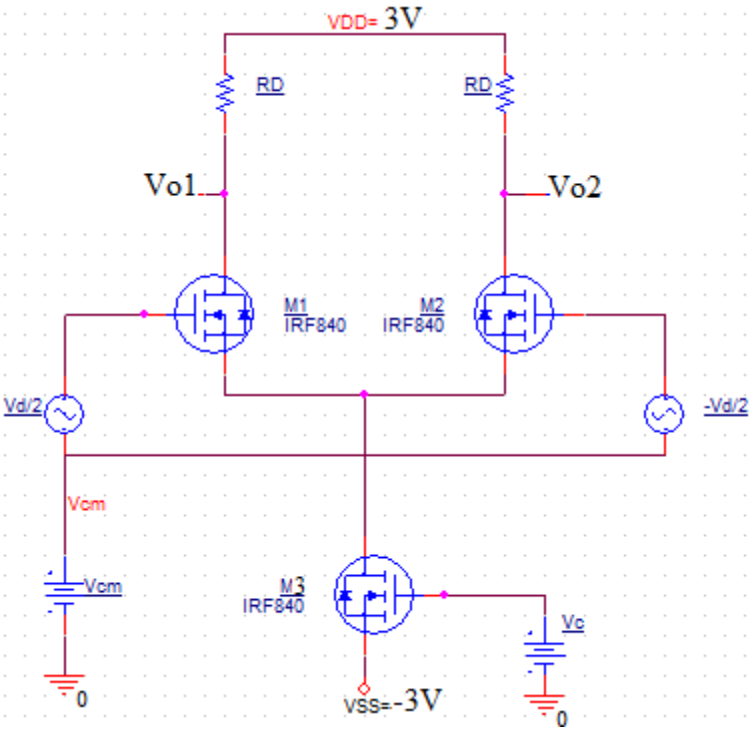


FIGURE 3.4: VGA USING CHOSEN COMPONENTS

The parameters of IRF840 power MOSFET, i.e threshold voltage V_{TN} and conduction parameter $Kn=\frac{1}{2}u_nC_{ox}\frac{W}{L}$ are determined as follows.

They can be determined by plotting the drain to source current, I_{DS} versus gate to source voltage V_{GS} graph and taking two points/values to determine the two parameters. To ensure that the MOSFET is in saturation, the drain to source voltage $V_{DS}\geq V_{GS}-V_{TN}$, one connects the get and drain so that $V_{DS}=V_{GS}$ as shown below.

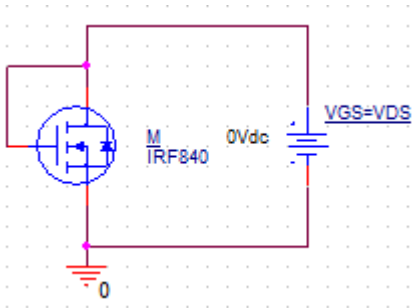


FIGURE 3.5 CIRCUIT FOR DETERMINING IRF840 PARAMETERS

I_{DS} versus V_{GS} graph is plotted in PSPICE using DC sweep analysis with V_{GS} as parameter varying from 2 to 5 volts as shown. Since the datasheet specifies the threshold voltage V_{TN}

for IRF840 is between 2 and 4 volts, VGS variation in DC sweep analysis is made to start at 2.

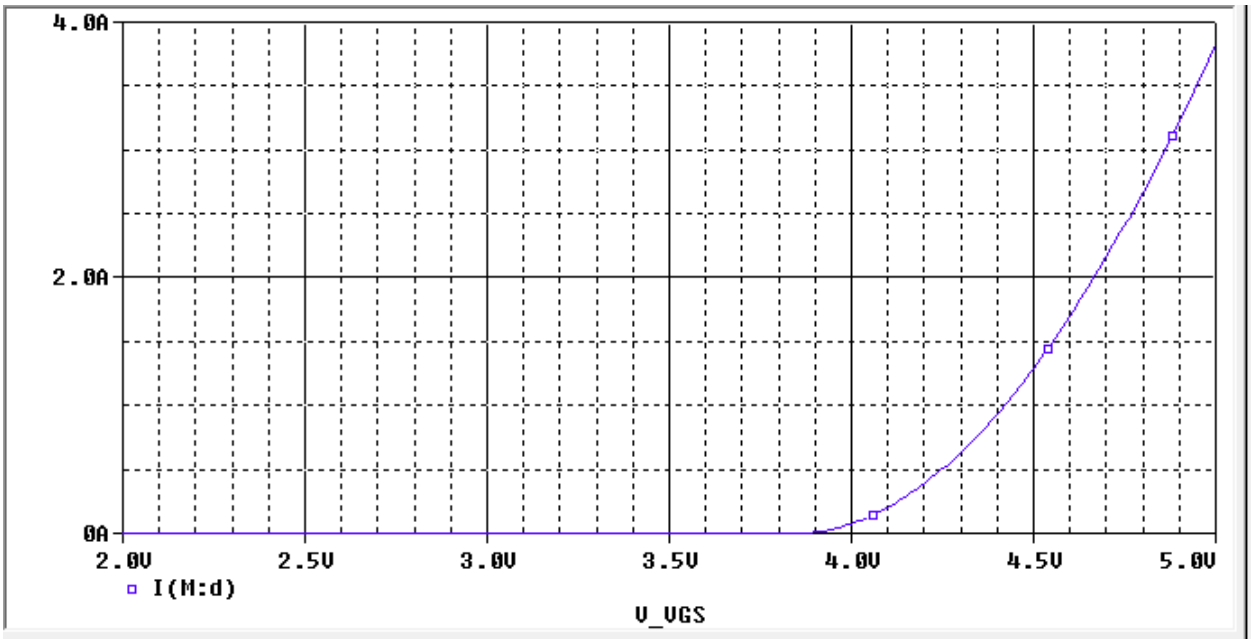


FIGURE 3.6: I_{DS} VS V_{GS} GRAPH FOR IRF840 POWER MOSFET

The following values of V_{GS} and I_{DS} in Table 3.1 are extracted from the above graph:

TABLE 3.1: SELECTED POINTS FROM IDS VS VGS GRAPH

V_{GS}	4V	4.5V
I_{DS}	73.27mA	1.2908

The drain to source current I_{DS} for saturation region is given by:

$$I_{DS}=Kn(V_{GS}-V_T) \quad \text{..... 3.1}$$

Where V_{TN} is the threshold voltage and $Kn=\frac{1}{2}u_nC_{ox}\frac{W}{L}$ is the conduction parameter. Accordingly, for two selected drain currents I_{DS1} and I_{DS2} as in Table 3.1 above:

$I_{DS1}=K_n(V_{GS1}-V_{TN})^2$

$I_{DS2}=K_n(V_{GS2}-V_{TN})^2$

Combining above two equations:

$$\sqrt{(\frac{IDS1}{IDS2})}=\frac{VGS1-VTN}{VGS2-VTN}$$

$$\sqrt{(\frac{73.27mA}{1.2908})}=\frac{4-VTN}{4.5-VTN}$$

Then the threshold voltage V_{TN} for the selected MOSFET (IRF840) will be:

V_{TN}=3.8439V

The conduction parameter K_n is found by substituting the threshold voltage into one of the above equations as follows:

$$K_n=\frac{IDS1}{(VGS1-VTN)2}=\frac{73.27mA}{(4-3.8439)2}=\mathbf{3.0069\frac{A}{V2}}$$

The single-ended voltage gain, Av in terms of control voltage, Vc is derived as follows,

Voltage gain for the right half circuit of the differential amplifier in Figure 3.4 above is given by:

$$\mathbf{A_v=g_mR_D} \quad \text{.....3.2}$$

The drain to source current, I_{DS} is also expressed as in equation 3.1 above:

$$I_{DS}=K_n(V_{GS2}-V_{TN})^2$$

The trans conductance parameter, g_m is expressed as:

$$g_m=\frac{\partial IDS}{\partial VGS}=2K_n(V_{GS}-V_{TN})=2K_n\sqrt{(\frac{IDS}{Kn})}$$

$$\mathbf{g_m=2\sqrt{(K_nI_{DS})}} \quad \text{.....3.3}$$

Combining equations 3.2 and 3.3, the single-ended voltage gain, Av of the differential amplifier is:

$$\mathbf{Av=2R_D\sqrt{(K_nI_{DS})}} \quad \text{.....3.4}$$

Expressed in terms of MOSFET 2 drain current, equation 3.4 will be:

$$Av=2R_D\sqrt{(K_nI_{DS2})}$$

But, $I_{DS1}=I_{DS2}=I_{DS3}/2$; therefore,

$$A_v=2R_D\sqrt{(K_n\frac{IDS3}{2})}$$

Substituting for I_{DS3}, the voltage gain will be:

$$A_v=R_D\sqrt{(2K_n * K_n(V_{GS3}-V_{TN})^2)}=R_D*K_n(V_{GS3}-V_{TN})\sqrt{2}$$

$$\mathbf{A_v=R_D*K_n(Vc-V_{SS}-V_{TN})\sqrt{2}} \quad \text{.....3.5}$$

The range of control voltage, Vc and the common mode input voltage Vcm are found as follows:

a) Control voltage, Vc range and selection

The maximum control voltage Vc(max) is determined from the maximum current that is chosen at M3. Assume the maximum drain to source current of M3, I_{DS3}(max)=2A and using equation 3.1 above:

$$I_{DS3}(max)=K_n(V_{GS3}(max)-V_{TN})^2$$

Substituting the above determined values for K_n and V_{TN}:

$2=3.0069(V_{GS3}(max)-3.8439)^2$, => $\mathbf{V_{GS3}(max)=4.6595.}$

$$V_{GS3}(max)=V_{G3(max)}-V_{s3}$$

But from the differential amplifier circuit in Figure 3.4 above:

V_{G3}=Vc and V_{S3}=V_{SS}; substituting this:

V_{GS3}(max)=Vc(max)-V_{SS}, => Vc(max)= V_{GS3}(max) + V_{SS}=4.6595 – 3=**1.6595**

The minimum value of control voltage, Vc is determined from the threshold voltage, i.e. the gate to source voltage of M3, V_{GS3}>V_{TN}.

V_{GS3}>V_{TN}

V_{G3}-V_{s3} > V_{TN}

Vc-V_{SS}> V_{TN}

Vc> V_{TN} + V_{SS}

Vc>3.8439-3 => Vc>0.8439

0.8439 < Vc≤ 1.6595

b) Selecting drain resistance, R_D

Nearly at the mid value of Vc, the gain is required to be unity so that the VGA can increase (when Av>1) or decrease (when Av<1) the amplitude level of incoming signals.

Vc(mid)=0.8439 + $\frac{(1.6595-0.8439)}{2}$= 1.25

From equation 3.5 above:

Av=R_D*K_n (Vc-V_{SS}-V_{TN})√2

1= R_D*3.0069(1.25+3-3.8439)√2

R_D=0.6

The nearest integer resistor value is, R_D=1Ω

Substituting this value of R_D and the other parameters in equation 3.5:

Av= 1*3.0069(Vc + 3 – 3.8439)√2

Av=4.2524(Vc-0.8439) ……3.6

c) Input common mode voltage, V_{cm} range and selection:

The minimum common mode voltage is determined such that transistor M3 should not leave saturation, i.e. the dain to source voltage of M3 should be, V_{DS3}>V_{GS3}-V_{TN}.

Using KVL in Figure 3.4:

$V_{cm(min)}$-V_{GS1}-V_{DSat3}-V_{SS}=0

$V_{cm(min)}$= V_{GS1} + V_{DSat3} + $V_{SS,}$ where V_{DSat3} is the saturation drain to source voltage for MOSFET 3 and given by V_{DSat3}=V_{GS3}-$V_{TN.}$

Fixing the maximum drain current of transistor 3 to 2A, I_{DS1}=I_{DS2}=$\frac{IDS3}{2}$=1A.

I_{DS3}(max)=K_n(V_{GS3}(max)-V_{TN})^2

Substituting the above determined values for K_n and V_{TN}:

$2=3.0069(V_{GS3}(max)-3.8439)^2$, => $V_{GS3}(max)=4.6595$.

$V_{DSat3}=V_{GS3}-V_{TN}=4.6595-3.8439= 0.8156$

$I_{DS1}=\frac{IDS3}{2}=K_n(V_{GS1}-V_{TN})^2$

Similarly, using the determined values for K_n and V_{TN}:

$1=3.0069(V_{GS1}-3.8439)^2$ => $V_{GS1}=4.4206$.

Using the determined values of V_{GS1} and V_{Dsat3}, the minimum common mode voltage, $V_{cm(min)}$ will be:

$V_{cm(min)}= V_{GS1} + V_{DSat3} + V_{SS}=4.4206 + 0.8156 -3=$**2.2362**

The maximum value of Vcm is determined such that transistor M1 should not leave saturation, i.e. the drain to source voltage for transistor 1 should be, $V_{DS1}>V_{GS1}-V_{TN}$ and using KVL in Figure 3.4 above:

$V_{cm(max)} - V_{GS1} + V_{DSat1} + I_{DS1}R_D – V_{DD}=0$

$V_{cm(max)} = V_{GS1} - V_{DSat1} - I_{DS1}R_D + V_{DD}$

The saturation drain to source voltage for transistor 1 is, $V_{DSat1}= V_{GS1}-V_{TN}=4.4206-3.8439=0.5767$. Using the determined values of drain resistance R_D and V_{DSat1}:

$V_{cm(max)}=4.4206 - 0.5767 - 1*1 + 3 = 5.8439$; therefore, the range of V_{cm} is,

$2.2362 \leq Vcm \leq 5.8439$

For maximum differential output voltage swing, the common mode input voltage V_{cm} should be selected in the mid value of range of V_{cm} above. Accordingly, $V_{cm}=4$ V is selected for design.

Since the amplifier circuit is multistage and so output of one stage is input to the next, the minimum output voltage of each stage should be same as the minimum common mode input voltage, Vcm, i.e. Vo(min)=Vcm(min)=2.2362.

The minimum output voltage $V_{o(min)}$ is also occurred when I_{DS} is maximum because $V_o=V_{DD} - I_{DS}R_D$.

Accordingly, $Vo1(min)= V_{DD} -I_{DS1(max)}R_D=3-I_{DS1(max)*1}$

$2.2362=3-I_{DS1(max)}=3- \frac{IDS3(max)}{2}$

$I_{DS3(max)}=1.5$

But, $I_{DS3(max)}=K_n(V_{GS3(max)} - V_{TN})^2= K_n(V_{G3(max)}-V_{S3}- V_{TN})^2 = K_n(V_{c(max)}-V_{SS}- V_{TN})^2 =3.0069(V_{c(max)} + 3 -3.8439)^2 = 1.5$. Then,

$V_{c(max)}$ = 1.55; consequently, range of the control voltage calculated in part (a) above is adjusted to the following value:

0.8439 < Vc≤ 1.55, out of this range, the following range of Vc is selected for design:

0.9 ≤ Vc ≤ 1.5

Simulating the VGA circuit in Figure 3.4 above in ORCAD PSPICE with the selected values and parameters, the following Table 3.2 is obtained. The test input to the VGA is 20mVpeak differential input voltage swinging on 4 V common mode input voltage as shown in Figure 3.7 below:

Table 3.2: Comparison of calculated and simulated values of VGA output

Vc (V)	Av=4.2524(Vc-0.8439)	Vop-p(calculated) (mV)	Vop-p(simulated) (mV)	Difference(error) (mV)
0.9	0.2386	9.54	7.75	1.79
1.0	0.6638	26.55	26.72	0.17
1.1	1.0890	43.56	44.52	0.96
1.2	1.5143	60.57	61.58	1.01
1.3	1.9395	77.58	77.9	0.32
1.4	2.3648	94.59	94.16	0.43
1.5	2.7900	111.60	109.71	1.89

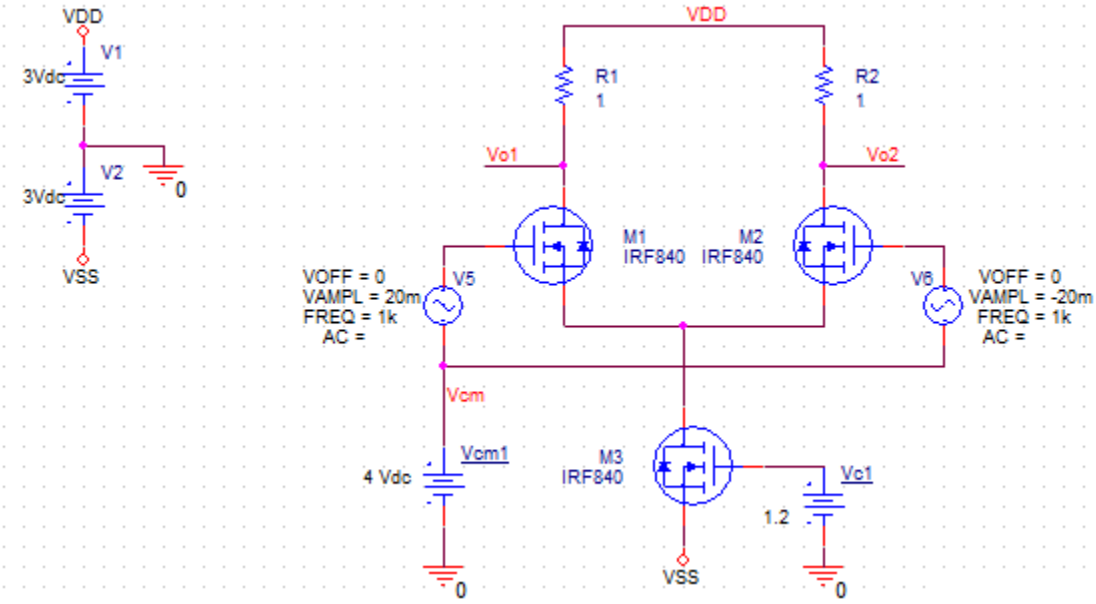


FIGURE 3.7: VGA WITH ITS VALUES AND BIASING SELECTED

As it can be observed from Table 3.2 above, the peak to peak differential output calculated and simulated are equivalent with a difference of small values occurred due to approximation when gain and parameters are calculated.

The range of control voltage Vc as determined above is, $0.9 \leq Vc \leq 1.5$. This has 20log (2.79/0.2386) =21dB gain range which shows the designed VGA is functioning well. Three such amplification stages are cascaded to increase the gain range to 3*21dB=63dB as shown in Figure 3.8 below.

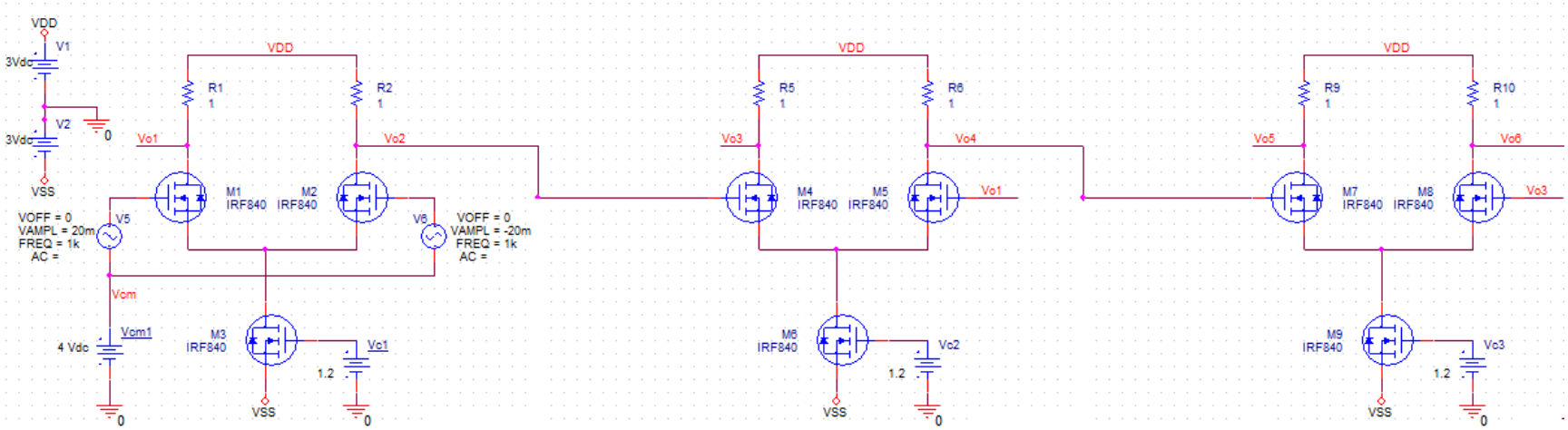


FIGURE 3.8: 3-TAGE VGA

At the end of the multistage amplifier, buffer (unity gain, high input impedance and low output impedance amplifier) is required so that it can derive low impedance load. As discussed in [18], the source follower exhibits a very high input impedance and a relatively low output impedance, thereby providing good buffering capability.

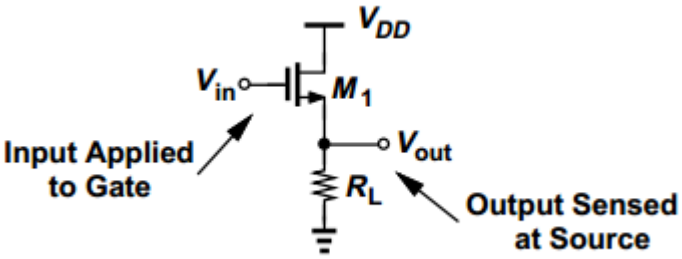


FIGURE 3.9: SOURCE FOLLOWER

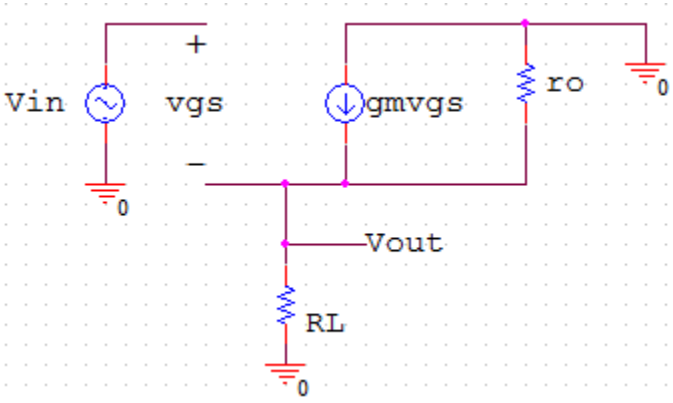


FIGURE 3.10: SMALL SIGNAL EQUIVALENT OF SOURCE FOLLOWER

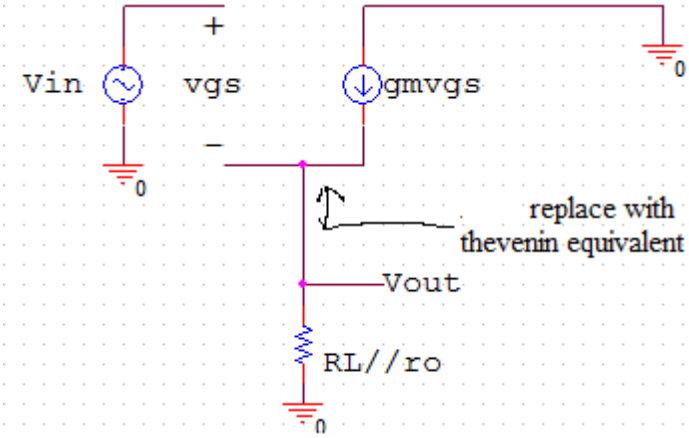


FIGURE 3.11: REPLACE THE NETWORK ABOVE VOUT WITH THEVENIN EQUIVALENT

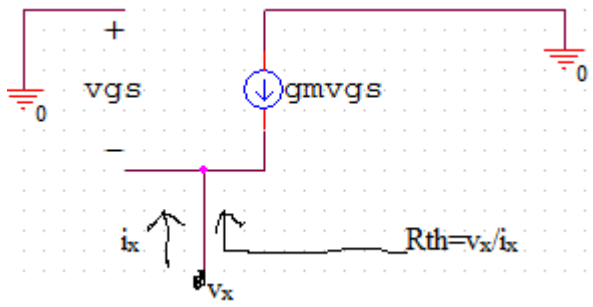


FIGURE 3.12: THEVENIN RESISTANCE, RTH DETERMINATION

Using KCL, $-i_x - g_m v_{gs}=0$

$-i_x + g_m v_x=0$

$Rth=v_x/i_x=1/g_m$

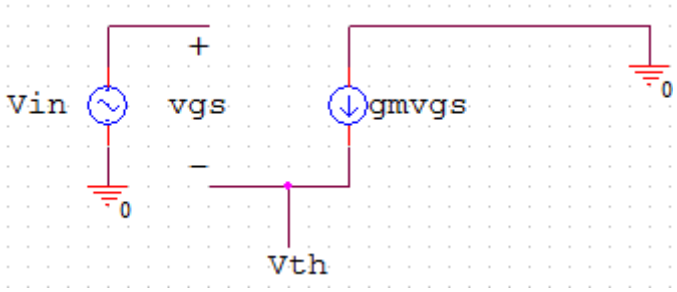


FIGURE 3.13: CIRCUIT TO FIND THEVENIN VOLTAGE, VTH

From KCL, $g_m v_{gs}=0 \Rightarrow v_{gs}=0$.

$v_{in}-v_{th}=v_{gs}=0 \Rightarrow v_{in}=v_{th}$

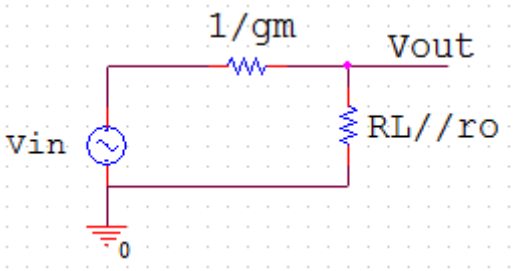


FIGURE 3.14:SIMPLIFIED SOURCE FOLLOWER CIRCUIT

Next, the gain and output resistance of the source follower circuit from Figure 3.14 will be obtained.

Applying KCL, $\frac{Vout}{RL//ro} + \frac{Vout-Vin}{1/gm} = 0$**……3.7**

Solving 3.9 for the voltage gain Av, equation 3.10 below is obtained.

Av=$\frac{Vout}{Vin} = \frac{gm(RL//ro)}{1+gm(RL//ro)}$**…… 3.8**

From equation 3.10, if RL//ro is >>1, then Av≈1 which has unity gain as buffer.

The output impedance, Zo is found by turning off independent sources, i.e Vin=0.

$Zo=R_L//r_o//1/g_m \approx 1/g_m$……..3.9

From equation 3.9, Zo is small as required, i.e the circuit can derive low impedance loads.

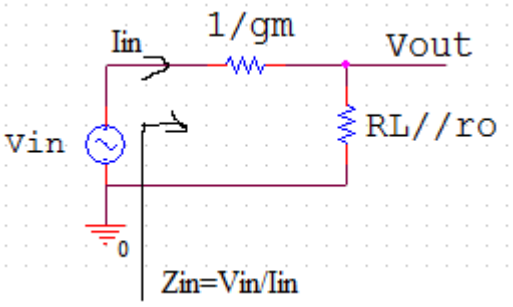


FIGURE 3.15: SOURCE FOLLOWER INPUT IMPEDANCE

The input impedance is the impedance seen from input side as shown in Figure 3.15 above.

Using KVL, Vin - I_{in}*1/g_m – I_{in}*(R_L//ro) = 0…….……3.10

Solving 3.12, $Z_{in} = \frac{Vin}{Iin}$ =1/g_m + R_L//ro ≈ R_L//ro ……..…..3.11

As can be observed from equation 3.11 above, the input impedance Z_{in} will be large if R_L is large. Therefore, if R_L is replaced with MOSFET current source M_2 which has infinite impedance r_{o2} in parallel as in Figure 3.16 below, a buffer with unity gain, high input impedance and low output impedance can be obtained.

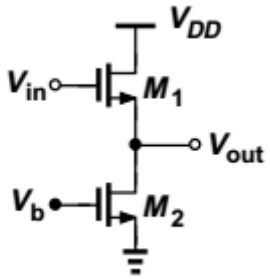


FIGURE 3.16: SOURCE FOLLOWER WITH CURRENT SOURCE LOAD

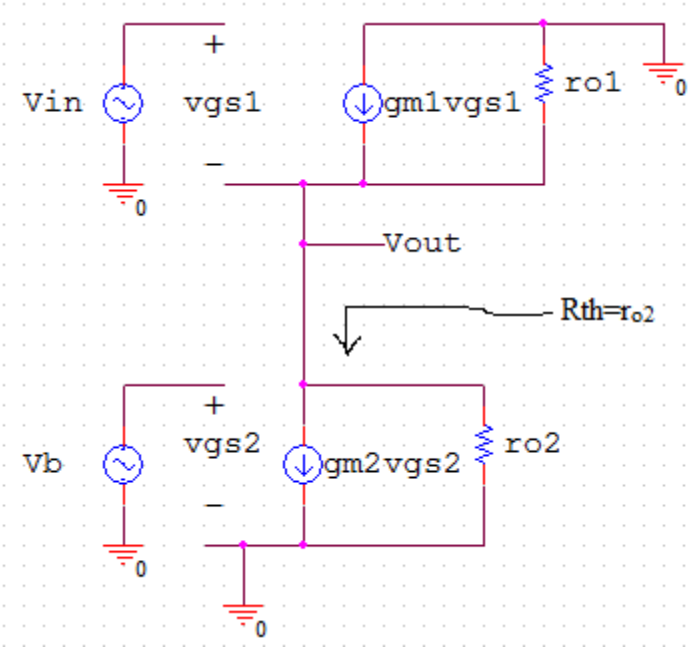


FIGURE 3.17: SMALL SIGNAL EQUIVALENT FOR THE SOURCE FOLLOWER WITH MOSFET CURRENT SOURCE AS LOAD.

As can be seen from Figure 3.17 above, R_L is replaced by r_{o2}. Therefore, the corresponding gain, input impedance and output impedance will be:

Replacing R_L with r_{o2} in equation 3.8 above, the voltage gain is,

$Av=\frac{Vout}{Vin} = \frac{gm1(ro2//ro1)}{1+gm1(ro2//ro1)} \approx 1$, as $r_{o2}//r_{o1} >>1$.

Also replacing R_L with r_{o2} in equation 3.11 above, the input impedance Z_{in} is,

$Z_{in} = \frac{Vin}{Iin} = 1/g_{m1} + r_{o2}//r_{o1} \approx r_{o2}//r_{o1}$ which is very large since M1 and M2 will be designed to be in saturation, i.e. current source. The advantage for large Z_{in} is that it will not affect the gain of previous circuits when cascaded with other amplifiers or circuits.

Similarly, replacing R_L with r_{o2} in equation 3.9,

$Zo=r_{o2}//r_{o1}//1/g_m \approx 1/g_m$ which is very small as required, i.e the circuit is capable of deriving low impedance loads. In other ways, the output voltage will be transferred to the next circuits with less effect, i.e. without being degraded by the output impedance.

Next, two source follower circuits are designed same as Figure 3.16 using IRF840 MOSFET to be connected to the output of the multistage amplifier in Figure 3.8 so that they can buffer the output.

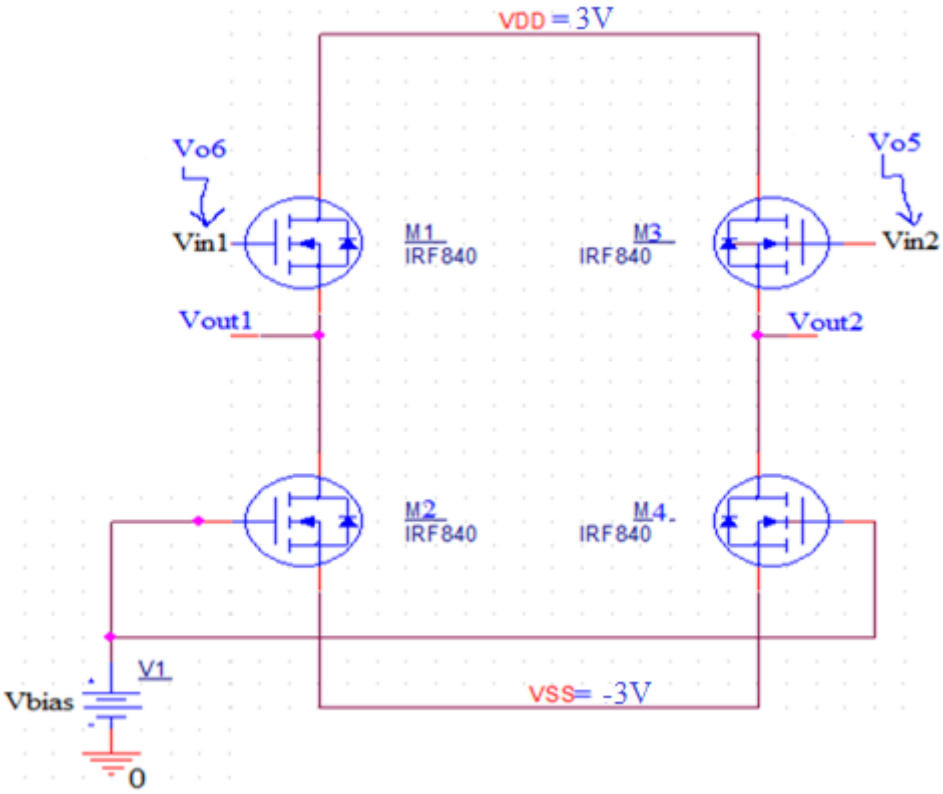


FIGURE 3.18: BUFFER CIRCUIT FOR THE DIFFERENTIAL PAIR VGA

The inputs to M1 and M3 are the outputs from the third VGA stage V_{o6} and V_{o5} respectively. Next, biasing voltage V_{bias} and common mode voltage V_{cm} are designed for the selected MOSFET IRF840.

The minimum bias, $V_{bias(min)}$ that can turn on the transistors is:

$V_{GS2}=V_{TN}$

$V_{G2}-V_{S2}=V_{TN}$

$V_{bias(min)}$ is connected to V_{G2} and V_{SS} is to V_{S2}. Substituting these values,

$V_{bias(min)} - V_{SS} = V_{TN}$, then the bias voltage will be,

$V_{bias(min)} = V_{TN} + V_{SS} = 3.8439 – 3= 0.8439$. Therefore,

Since $V_{bias}> 0.8439$, $V_{bias}=1V$ is selected for the design of the buffer.

Accordingly, the gate to source voltage of MOSFET M2, V_{GS2} is,

$V_{GS2} = V_{bias} - V_{SS} = 1+3=4$ V.

Since same drain to source current, I_{DS} flows in all transistors, $V_{GS1}=V_{GS2}=V_{GS3} = V_{GS4}= 4V$.

$I_{DS1}=K_n(V_{GS1}-V_{TN})^2$

The overdrive voltage, $V_{DSat}=V_{GS}-V_{TN}=4-3.8439= 0.8439 =V_{DSat1}=V_{DSat2}=V_{DSat3}=V_{DSat3}$.

The minimum value of input common mode voltage V_{cmb} for the buffer is determined such that M2 should not leave saturation. Applying KVL:

$V_{cmb} - V_{GS1} - V_{DSat2} - V_{SS} = 0$

$V_{cmb} = V_{GS1} + V_{DSat2} + V_{SS} = 4 + 0.8439 - 3 = 1.8439$

The maximum value of Vcmb is determined such that M1 should not leave saturation. Applying KVL:

$V_{cmb} - V_{GS1} + V_{DSat1} - V_{DD} = 0$

$V_{cmb} = V_{GS1} - V_{DSat1} + V_{DD} = 4 - 0.8439 + 3 = 6.1561$; therefore, the range of V_{cmb} is,

$1.8439 \leq Vcmb \leq 6.1561$

Since output of one stage is input to the next in multistage amplification, the minimum input common mode voltage Vcm is same as the minimum output Vo. The minimum Vcm for one variable gain amplifier as found in the previous section is Vcm =2.2362 V which is same as the minimum DC output. Therefore, as seen from range of Vcmb for the buffer above, the designed buffer can accommodate the minimum DC output of the amplifier behind it. The cascaded 3-stage VGA along with the designed buffer is shown in Figure 3.19 below:

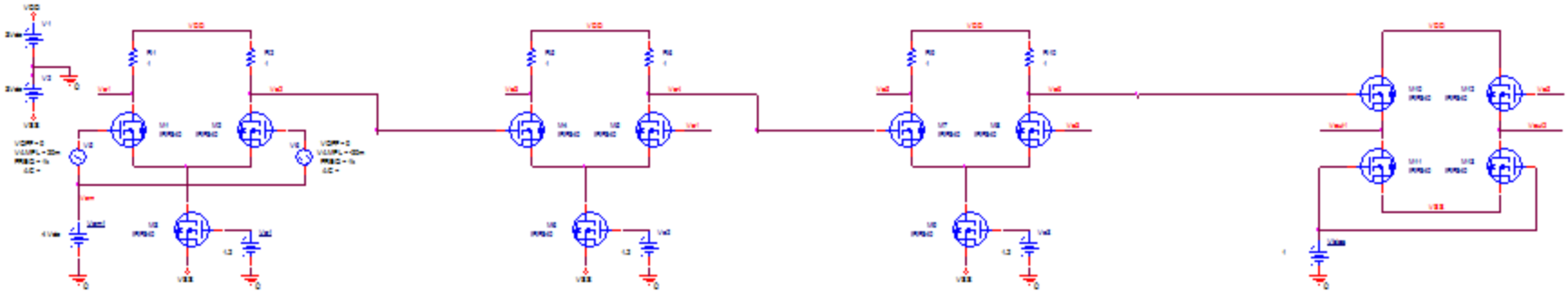


FIGURE 3.19: 3-STAGE VGA WITH BUFFER CONNECTED TO OUTPUT OF LAST STAGE

3.2 INSTRUMENTATION AMPLIFIER

An instrumentation amplifier is a differential voltage-gain device that amplifies the difference between the voltages existing at its two input terminals. The main purpose of an instrumentation amplifier is to amplify small signals that may be riding on large common mode voltages [19].To measure the differential ac output and remove the common mode dc output in the designed VGA, an instrumentation amplifier is needed.

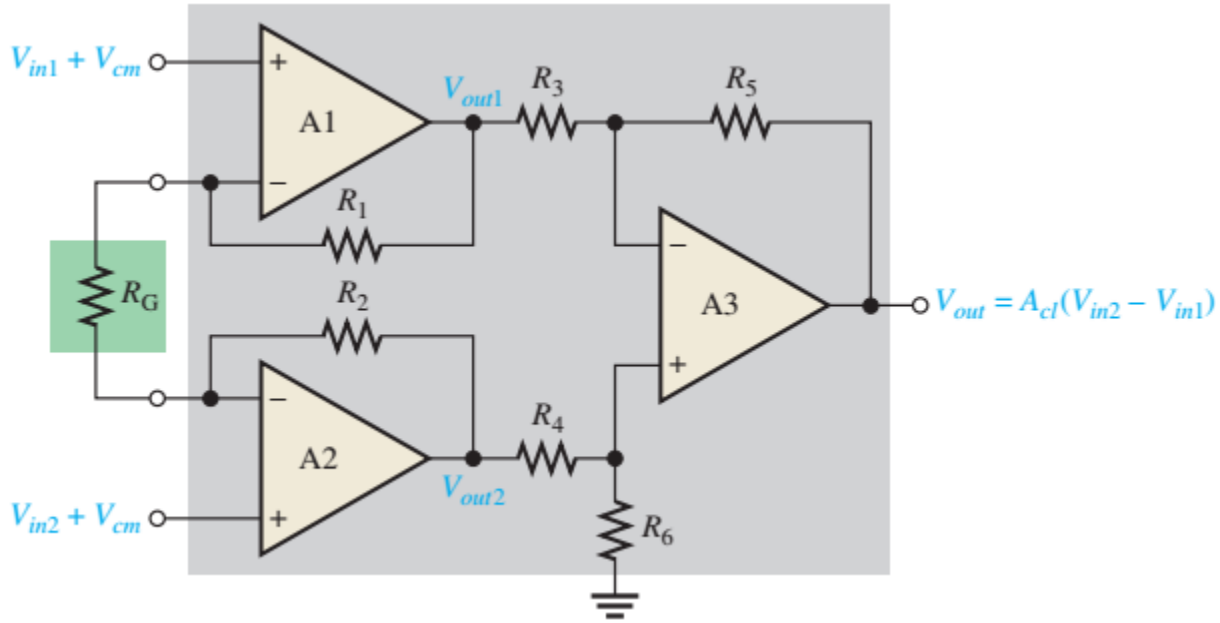


FIGURE 3.20: INSTRUMENTATION AMPLIFIER

Next is derived the closed loop gain A_{cl} considering $R_1=R_2=R_3=R_4=R_5=R_6=R$ in Figure 3.20 above. Let $V_1= Vin_1 + V_{cm}$ and $V_2= V_{in2} + V_{cm}$. Since the circuit is linear, superposition theorem can be used to derive the gain.

- V_{out} due to only V_1 by making $V_2=0$,

  In this case, OPAMP1 is acting as non-inverting and OPAMP2 as inverting amplifiers.

  $V'_{out1} = (1 + \frac{R1}{RG})V_1 = (1 + \frac{R}{RG})V_1$

  $V'_{out2} = - \frac{R2}{RG}V_2 = - \frac{R}{RG}V_1$

- V_{out} due to only V_2 by making $V_1=0$.

  In this case, OPAMP1 is acting as inverting and OPAMP2 as non-inverting amplifiers.

  $V''out1 = - \frac{R1}{RG}V_2 = - \frac{R}{RG}V_2$

  $V''_{out2} = (1 + \frac{R2}{RG})V_2 = (1 + \frac{R}{RG})V_2$

$V_{out1}=V'_{out1}+V''_{out1}=(1+\frac{R}{RG})V_1-\frac{R}{RG}V_2$ 3.12

$V_{out2}=V'_{out2}+V''_{out2}=(1+\frac{R}{RG})V_2-\frac{R}{RG}V_1$ 3.13

- OPAMP3 is a differential amplifier circuit and its gain is found similarly using superposition:

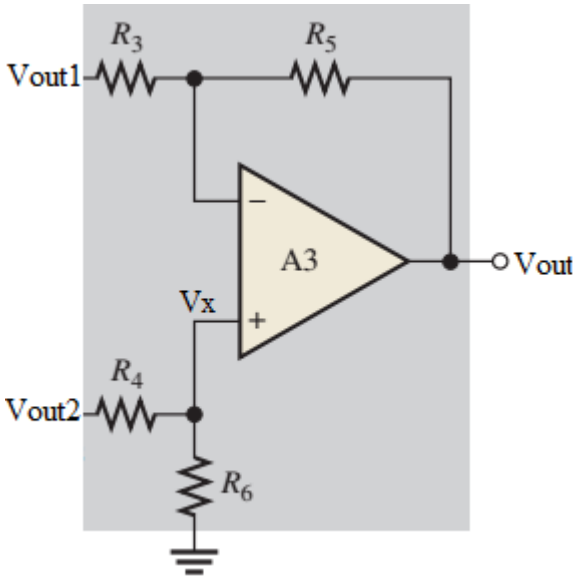


FIGURE 3.21: DIFFERENCE AMPLIFIER

- Due to V_{out1} in which case OPAMP3 acting as inverting OPAMP:

  $V'_{out}=-\frac{R5}{R3}$ Vout1

- Due to Vout2 in which case OPAMP3 acting as non-inverting OPAMP.

  Using KCL at node Vx in Figure 3.21:

  $V''_{out}=(1+\frac{R5}{R3})Vx$

  $\frac{Vx}{R6}+\frac{Vx-Vout2}{R4}=0$

  $Vx=\frac{R6}{R4+R6}Vout2$

  $V''_{out}=(\frac{R3+R5}{R3})\frac{R6}{R4+R6}V_{out2}$

$V_{out}=V'_{out}+V''_{out}=(\frac{R3+R5}{R3})\frac{R6}{R4+R6}V_{out2}-\frac{R5}{R3}$ V_{out1}.....3.14

If $R_3=R_4=R_5=R_6=R$, $V_{out}=V_{out2}-V_{out1}$ 3.15

Combining equations 3.12, 3.13, 3.14 and 3.15, the output of the instrumentation amplifier is,

$V_{out}=(1+\frac{2R}{RG})V_{out2}-(1+\frac{2R}{RG})V_{out1}=(1+\frac{2R}{RG})(V_{out2}-V_{out1})=(1+\frac{2R}{RG})(V_{cm}+V_{in2}-V_{cm}-V_{in1})$

Solving the above equation, the closed loop gain, A_{cl} is,

$A_{cl}=\frac{Vout}{Vin2-Vin1}=1+\frac{2R}{RG}$ 3.16

Using equation 3.16, one can select the ratio of R and R_G based on the requirement of closed loop gain, A_{cl}.

From the previous section, the range of common mode voltage V_{cm} is $2.2362 \leq V_{cm} \leq 5.8439$ and from this range V_{cm}=4 V is selected for the design; consequently, the maximum allowable differential ac voltage swing Vd is, Vd=4-2.2362=1.7638 Vpeak.

Since there are three stages, if the maximum differential input at the third stage is 1.7638 Vpeak, the maximum differential input at first stage $v_{d1max(peak)}$ is determined as follows,

The maximum control voltage that is determined in previous section is Vc=1.5. The maximum single ended gain, Av=2.79, for the three stages, the total gain A_{tot}=2.79*2.79*2.79= 21.718

Atot * $V_{d1max\ (peak)}$ = 1.7638 V=1763.8 mV

$V_{d1max\ (peak)} = \frac{1763.8\ mV}{21.718} \approx$ **81 mVpeak**

Therefore, for the design in this thesis work the maximum amplitude value allowed at input of the multistage amplifier is 81 mVpeak.

uA741 OPAMP is chosen to design the instrumentation amplifier. From the datasheet of uA741, the maximum input swing is ±15 and the maximum output swing ±10.

The maximum single-ended output at the last stage is,

Vo(max)=Vodc(max) + Vd = Vodc(max) + 1.7638 Vpeak.

But, the maximum dc output occurs when the drain to source current I_{DS} is minimum, i.e. when I_{DS} =0 A.

Accordingly, $Vodc(max)=V_{DD}-I_{DS(min)}R_D= V_{DD} - 0*1= V_{DD}$=3 V. Therefore,

Vo(max)=3+1.7638=4.7638 V which is within the input range of the OPAMP (±15).

Assuming the closed loop gain of the instrumentation amplifier, A_{cl} = 2, i.e.

$$\mathbf{A_{cl}= \left(1 + \frac{2R}{RG}\right) =2} \quad \textbf{.....3.17}$$

To fulfil equation 3.17, $\frac{R}{RG}$=0.5.

Let $V_{out(max)}$ is the peak value of the output of the instrumentation amplifier at third/last stage.

$V_{out(max)}$=2(Vin2-Vin1) =2(1.7638 - (-1.7638)) = 4*1.7638 =7.0552 which is within range of output of the OPAMP (±10). Note that the single-ended outputs of the differential amplifier are same in magnitude and opposite in phase. Therefore, at output of the instrumentation amplifier, the single-ended output of the differential amplifier is multiplied by a factor of 4.

The instrumentation amplifier has three advantages in the design:

1) Removes the common mode signal
2) Combines the two single-ended outputs and provides differential output
3) Amplifies the differential output which makes small signals to be readable by the ADCs with better resolution.

Based on above calculations, the following components are used to design the instrumentation amplifier:

- uA741 OPAMP
- R=1k and R_G=2k so that equation 3.17 will be fulfilled.

The designed instrumentation amplifier is shown in Figure 3.22 below. Three such amplifiers will be connected to each VGA output in the 3-stage VGA designed as in Figure 3.19 above.

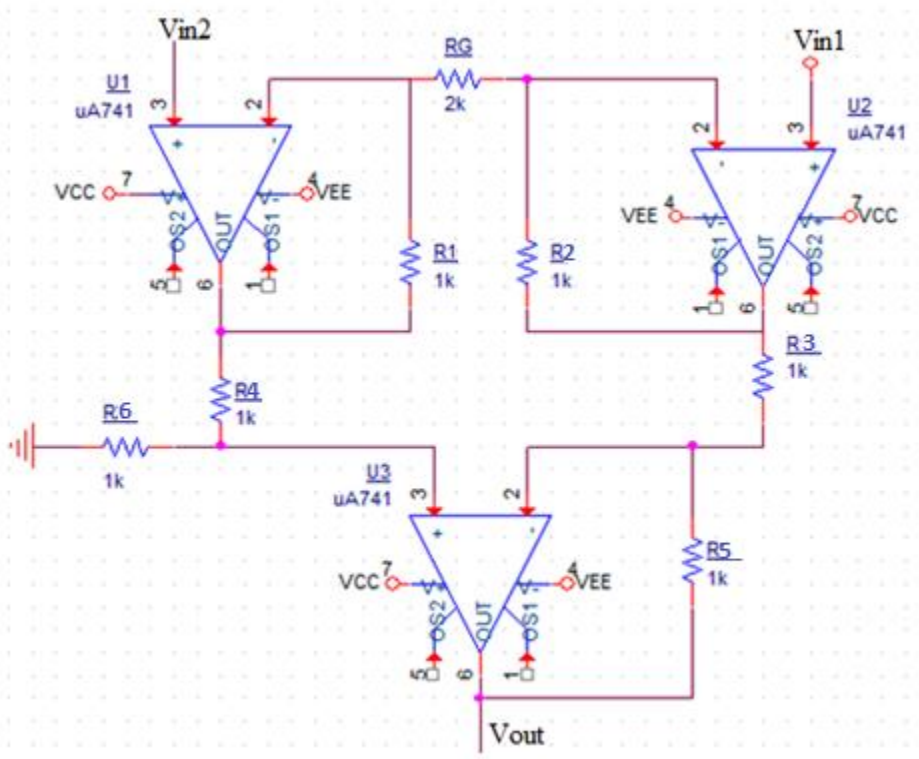


FIGURE 3.22: DESIGNED INSTRUMENTATION AMPLIFIER

3.3 PEAK DETECTOR CIRCUIT

As discussed in [19], the peak detector circuit is used to detect the peak of the input voltage and store that peak voltage on a capacitor. It can be used to detect and store the maximum value of an AC voltage; this value can then be measured at the output with a voltmeter or recording device.

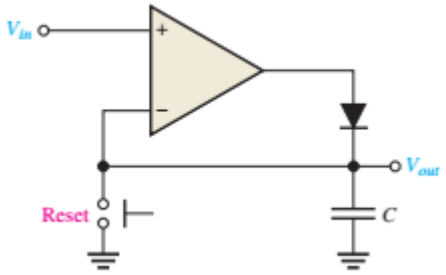


FIGURE 3.23: PEAK DETECTOR

Selecting the necessary components, i.e uA741 OPAMP, 1N4007 diode, capacitor, IRF840 MOSFET for switching and a digital pulse to turn on and off the MOSFET, the peak detector circuit shown in Figure 3.23 above is designed in ORCAD PSpice as shown in Figure 3.24 below. The purpose of the switch is to reset the value in the capacitor so that it can hold new values.

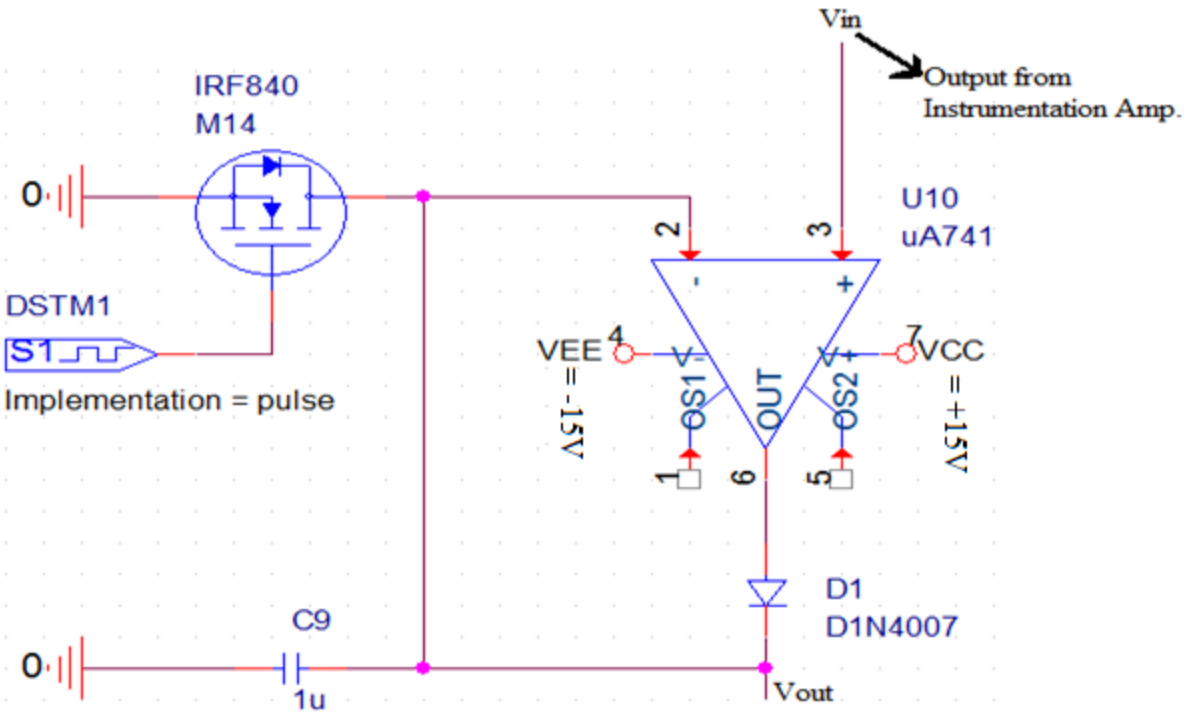


FIGURE 3.24: PEAK DETECTOR CIRCUIT AS DRAWN IN PSPICE

The pulse value should be greater than threshold value of the MOSFET. In this case, it is selected to be 5V and the period is 1ms which is same as the period of the input signal. The comparison of the input to the peak detector and the corresponding detected peak value is shown in Figure 3.25 below. The output is evaluated at first stage for Vc=1.3V, and Vd=20mVpeak.

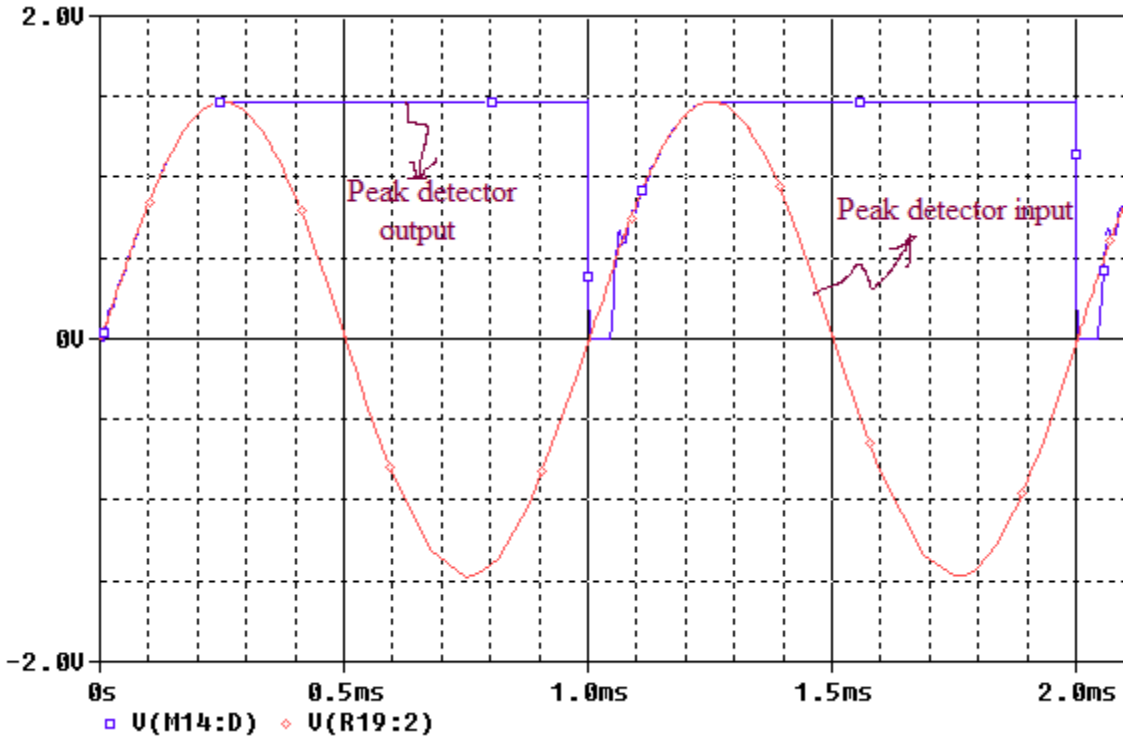


FIGURE 3.25: COMPARISON OF THE OUTPUT OF THE PEAK DETECTOR AND THE CORRESPONDING SINUSOIDAL INPUT

The analog part of the design as implemented and tested in Orcad Pspice is shown in Figure 3.26 below.

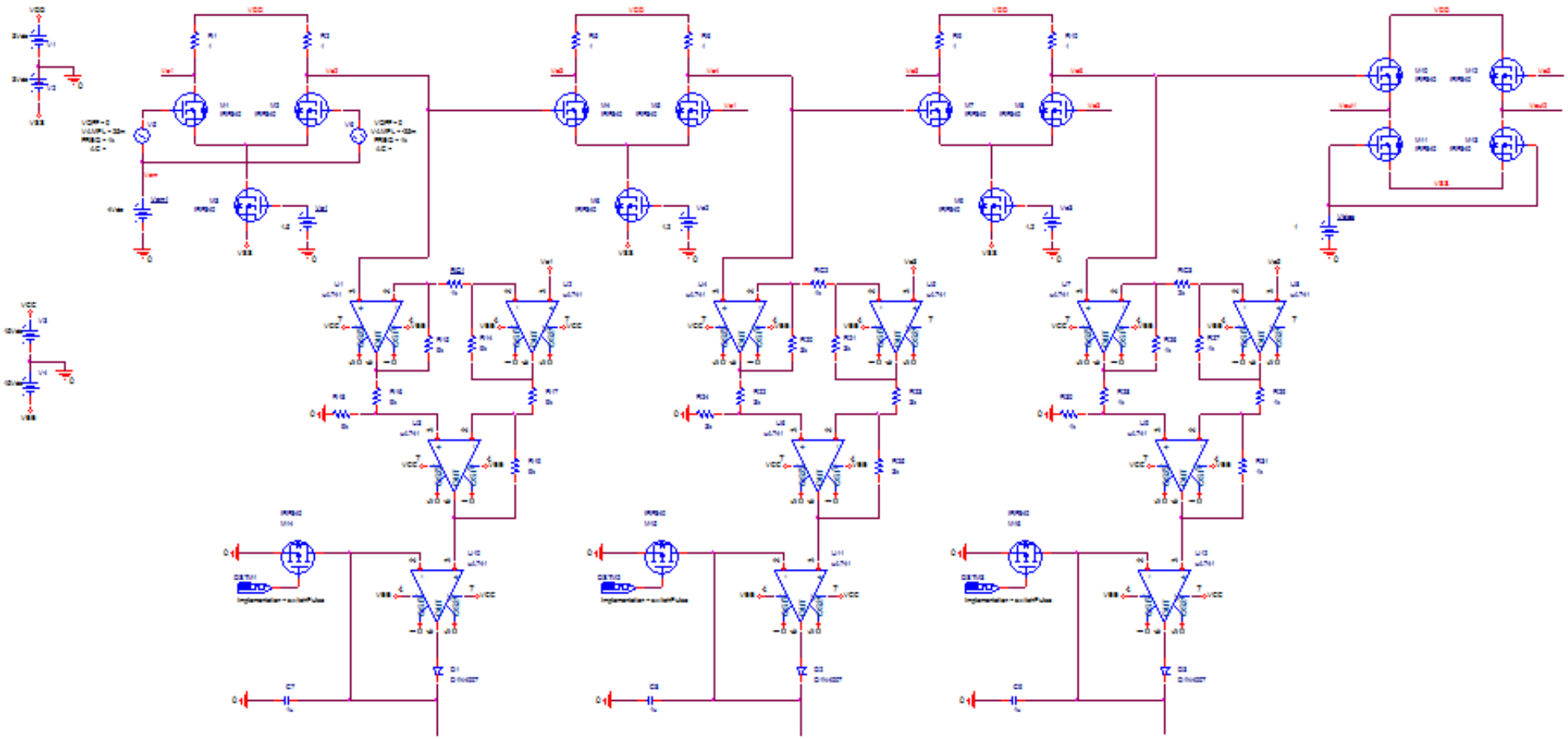


FIGURE 3.26: ANALOG PART OF THE SYSTEM DESIGN

3.4 ANALOG TO DIGITAL AND DIGITAL TO ANALOG CONVERTERS

One of the most important functions in signal processing is the conversion between analog and digital signals. In order to compare the peak amplitude value with the reference value in FPGA so that to generate control voltage, one need to convert the peak value which is analog to digital equivalent. For such purpose, Analog to Digital Converter (ADC) is needed. After generating the control voltage in FPGA, it is needed to convert it to analog equivalent value to be fed back to analog circuits and for such purpose Digital to Analog Converter (DAC) is required. Discussion about ADC and DAC will be followed and it is based on reference [20].

3.4.1 ANALOG TO DIGITAL CONVERTER (ADC)

ADC is a sampled-data circuit which converts analog signal to equivalent digital signal by sampling. The time that the sampled analog signal is converted to equivalent digital one is called conversion time. The conversion is accomplished by quantization step. The nature of a quantizer is to segment the reference into subranges. Typically, there are 2^N subranges, where N is the number of bits of the digital output code. The quantization step finds the subrange that corresponds to the sampled analog input. Thus, within the conversion time, a sampled analog input signal is converted to an equivalent digital output code.

The input of an ADC can have any value between 0 and V_{REF}. The resolution of an ADC is the smallest analog change that can be distinguished by an ADC. It is typically given by the number of bits N, where the converter has 2^N output possible states. The output of the ADC in decimal equivalent is:

Vout= $\frac{Vin}{Vref}$ *** (2^N-1)……..3.18**

The output of the ADC is digital code and if one needs to convert it back to analog for comparison purposes, the DAC output formula given by equation 3.21 below, i.e.,

$$V_{analog} = V_{REF}\left(\frac{b_{N-1}}{2^1} + \frac{b_{N-2}}{2^2} + \ldots + \frac{b1}{2^{N-1}} + \frac{b0}{2^N}\right)$$

N is the total number of bits and b_i is the i^{th}-bit coefficient and is either 0 or 1.

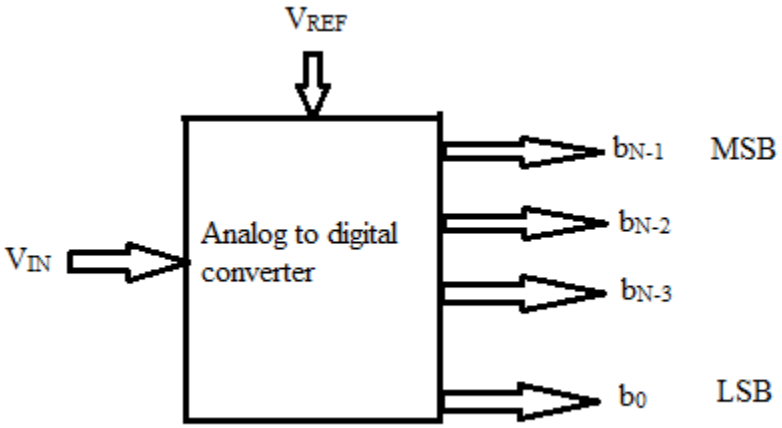


FIGURE 3.27: ADC BLOCK DIAGRAM

The sampling frequency is selected to be 10kHz in the design. The frequency of the input signal is 1kHz and then 10kHz≥2*1kHz which fulfils Nyquist criterion and sampling theorem [21].

3.4.2 DIGITAL TO ANALOG CONVERTER (DAC)

The input to DAC is a digital word consisting of parallel binary signals that are generated from signal processing system, in case of this thesis work, FPGA. These parallel binary signals are converted to equivalent analog signal by scaling a reference.

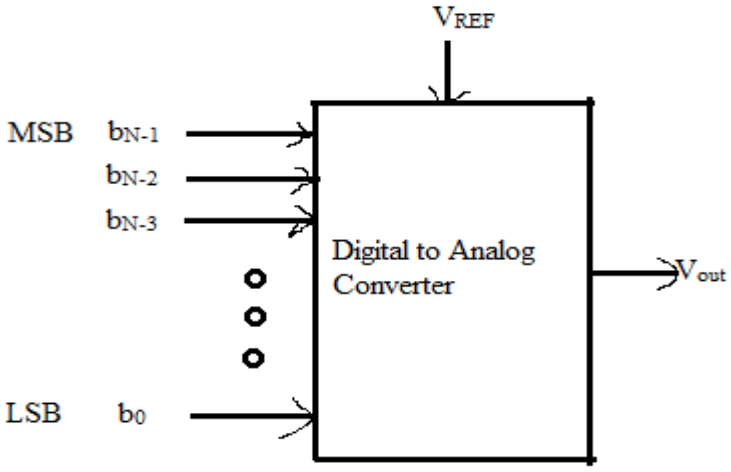


FIGURE 3.28: DAC BLOCK DIAGRAM

As shown in Figure 3.28 above, a DAC consists of a digital word of N-bits ($b_{N-1,}$ b_{N-2}, …, b_1, b_0) and a reference voltage V_{REF}. b_{N-1} is called the most significant bit (MSB) and b_0 is called the least significant bit (LSB). The voltage output V_{out} can be expressed as:

$$V_{out} = KV_{REF}D \quad \text{.....3.19}$$

Where K is the scaling factor and D is the digital word given us,

$$D = \frac{b_{N-1}}{2^1} + \frac{b_{N-2}}{2^2} + \ldots + \frac{b1}{2^{N-1}} + \frac{b0}{2^N} \quad \text{.....3.20}$$

The scaling factor K is considered to be 1 unless stated. Combining equations 3.19 and 3.20,

$$V_{out} = V_{REF}\left(\frac{b_{N-1}}{2^1} + \frac{b_{N-2}}{2^2} + \ldots + \frac{b1}{2^{N-1}} + \frac{b0}{2^N}\right) \quad \text{.....3.21}$$

N is the total number of bits and b_i is the i^{th}-bit coefficient and is either 0 or 1.

For implementation of this thesis work, the built-in ADCs and DACs in Xilinx Spartan 3E FPGA are used.

3.5 FIELD PROGRAMMABLE GATE ARRAY (FPGA)

In the computer and electronics world, two different ways of performing computation are used, hardware and software. Computer hardware, such as application-specific integrated circuits (ASICs), provides highly optimized resources for quickly performing critical tasks, but it is permanently configured to only one application via a multimillion-dollar design and fabrication effort. Computer software provides the flexibility to change applications and perform a huge number of different tasks, but is orders of magnitude worse than ASIC implementations in terms of performance, silicon area efficiency, and power usage. Field-programmable gate arrays (FPGAs) are truly revolutionary devices that blend the benefits of both hardware and software. They implement circuits just like hardware, providing huge power, area, and performance benefits over software, yet can be reprogrammed cheaply and easily to implement a wide range of tasks [22].

The purpose of the FPGA in this thesis work is to compute control voltage Vc. The FPGA is configured using VHDL code; however, before writing the actual VHDL code, an algorithm which is one of the problem-solving techniques in programming is developed. Next, state machine and VHDL code are developed respectively.

3.5.1 ALGORITHM

The variables that are used in the flow chart algorithm are:

Vc1, Vc2, Vc3: the control voltages for VGA1, VGA2 and VGA3 respectively

Vc10, Vc20, Vc30: the initial values of the control voltages

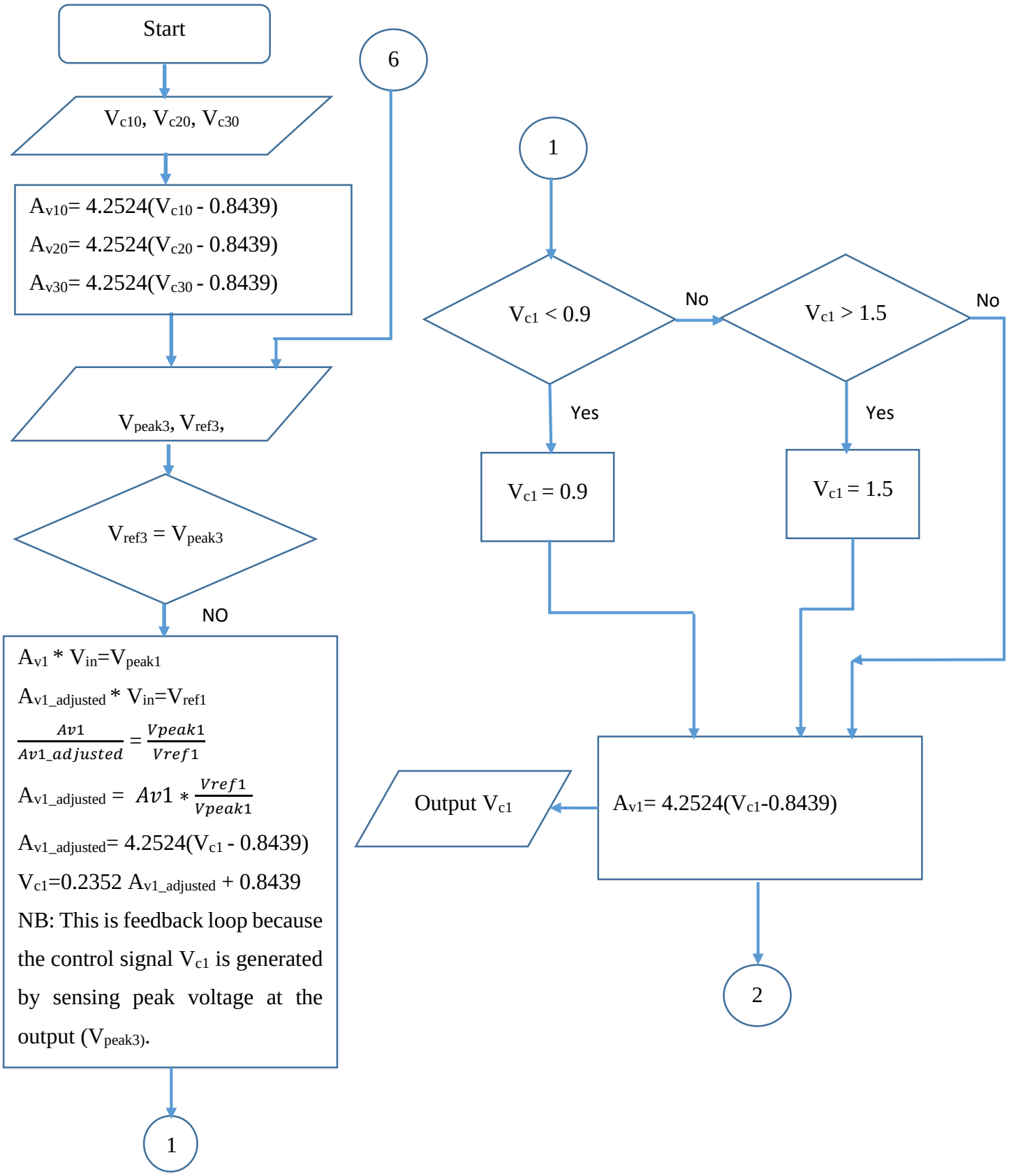
Vref1, Vref2, Vref3: the desired peak voltage outputs for VGA1, VGA2 and VGA3 respectively.

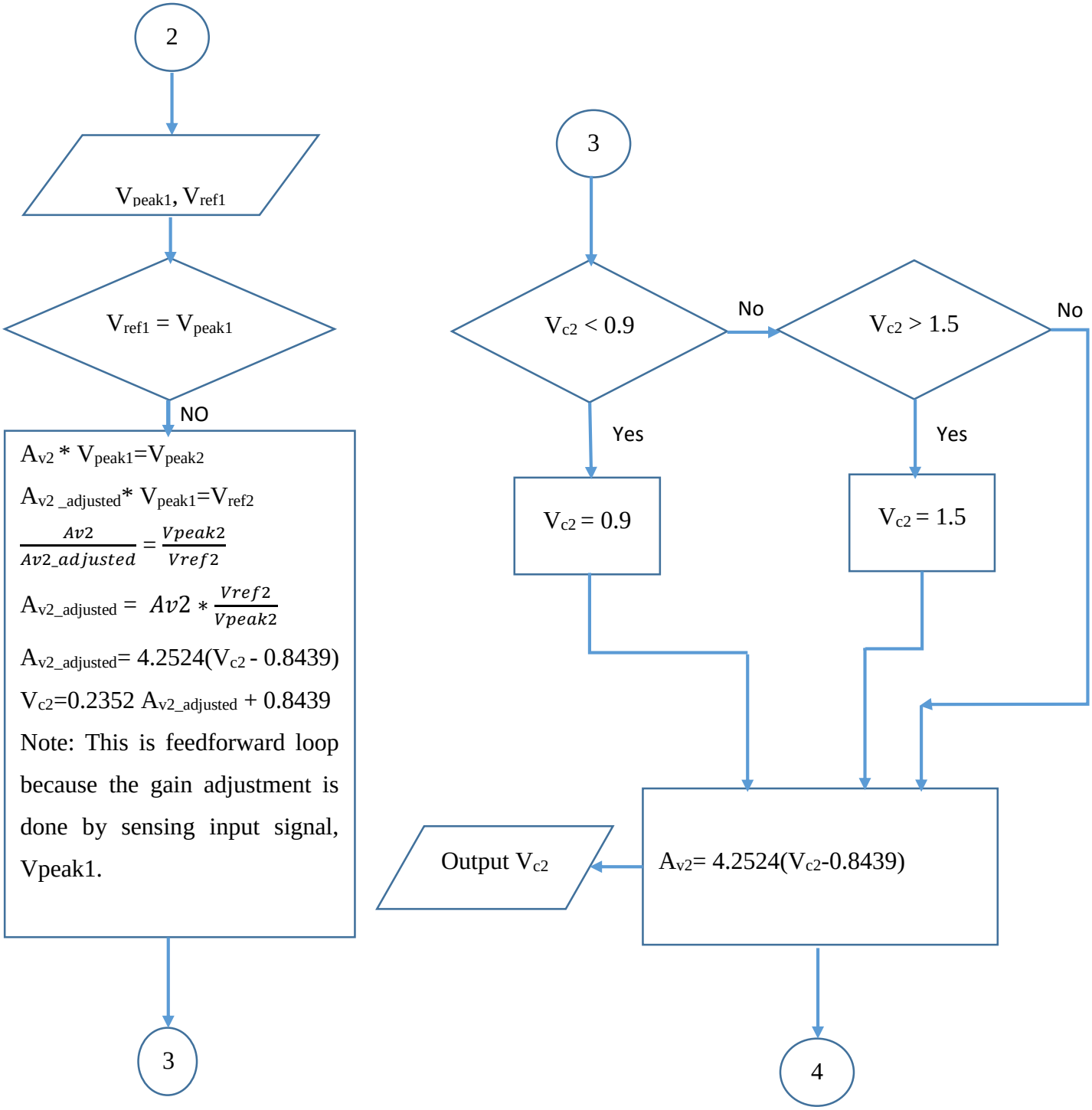
Vpeak1, Vpeak2 and Vpeak3: the sensed single-ended voltage outputs for VGA1, VGA2 and VGA3 respectively. The sensed peak voltage value at output of each stage is less than the output voltage of the peak detector by a factor of 4 due to the gain of the instrumentation amplifier.

Av1, Av2, Av3: the voltage gains for VGA1, VGA2 and VGA3 respectively.

Av10, Av20, Av30: the initial voltage gains at the reference inputs and outputs for VGA1, VGA2 and VGA3 respectively.

$A_{v1\_adjusted}$, $A_{v2\_adjusted}$, $A_{v3\_adjusted}$: the adjusted values of the gains for VGA1, VGA2 and VGA3 respectively.


Start
V_{c10}, V_{c20}, V_{c30}
A_{v10}= 4.2524(V_{c10} - 0.8439)
A_{v20}= 4.2524(V_{c20} - 0.8439)
A_{v30}= 4.2524(V_{c30} - 0.8439)
V_{peak3}, V_{ref3},
V_{ref3} = V_{peak3}
NO
A_{v1} * V_{in}=V_{peak1}
$A_{v1\_adjusted}$ * V_{in}=V_{ref1}
$\frac{Av1}{Av1\_adjusted} = \frac{Vpeak1}{Vref1}$
$A_{v1\_adjusted}$ = $Av1 * \frac{Vref1}{Vpeak1}$
$A_{v1\_adjusted}$= 4.2524(V_{c1} - 0.8439)
V_{c1}=0.2352 $A_{v1\_adjusted}$ + 0.8439
NB: This is feedback loop because the control signal V_{c1} is generated by sensing peak voltage at the output (V_{peak3}).
1
6
1
V_{c1} < 0.9
No
V_{c1} > 1.5
No
Yes
Yes
V_{c1} = 0.9
V_{c1} = 1.5
Output V_{c1}
A_{v1}= 4.2524(V_{c1}-0.8439)
2


2
V_{peak1}, V_{ref1}
$V_{ref1} = V_{peak1}$
NO
$A_{v2} * V_{peak1} = V_{peak2}$
$A_{v2\_adjusted} * V_{peak1} = V_{ref2}$
$\frac{Av2}{Av2\_adjusted} = \frac{Vpeak2}{Vref2}$
$A_{v2\_adjusted} = Av2 * \frac{Vref2}{Vpeak2}$
$A_{v2\_adjusted} = 4.2524(V_{c2} - 0.8439)$
$V_{c2} = 0.2352\ A_{v2\_adjusted} + 0.8439$
Note: This is feedforward loop because the gain adjustment is done by sensing input signal, Vpeak1.
3
3
$V_{c2} < 0.9$
No
$V_{c2} > 1.5$
No
Yes
Yes
$V_{c2} = 0.9$
$V_{c2} = 1.5$
Output V_{c2}
$A_{v2} = 4.2524(V_{c2} - 0.8439)$
4

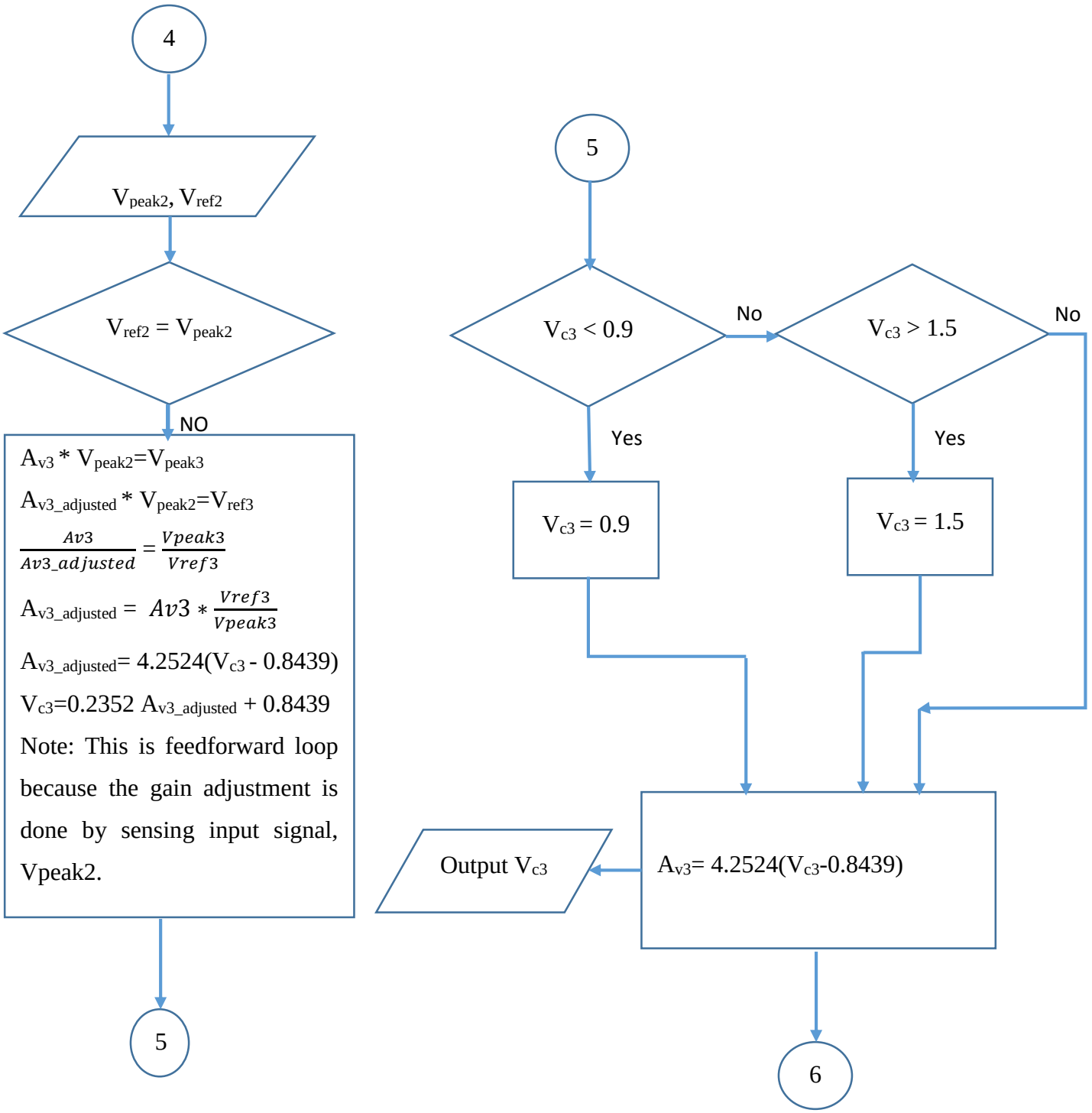


FIGURE 3.29: FLOW CHART ALGORITHM TO GENERATE CONTROL VOLTAGE, V_c

3.5.2 State Machine

The algorithm should be changed to state machine and then to VHDL code for digital hardware implementation. The concept for design of state machine and VHDL code that are used here is based on reference [23].

Let x is the output of a comparator for comparison of Vref and Vpeak. x=1 when Vref=Vpeak and 0 otherwise. The machine starts at the output of the last stage, i.e x=1 when Vref3=Vpeak3 and 0 otherwise. Let that initial state is S0. The machine generates control voltage Vc1 to adjust gain of VGA1. Note that this is feedback loop because the machine senses the output, generates Vc1 and fed it back to the first stage.

Then the machine proceeds to the output of the first stage, i.e x=1 when Vref1=Vpeak1 and 0 otherwise. Let that state is S1. The machine generates control voltage Vc2 to adjust gain of VGA2. Note that this is feedforward loop because the machine senses the input of VGA2, generates Vc2 and fed it forward.

Again, it proceeds to the output of the second stage, i.e x=1 when Vref2=Vpeak2 and 0 otherwise. Let that state is S2. The machine generates control voltage Vc3 to adjust gain of VGA3. Note that this is feedforward loop because the machine senses the input of VGA3, generates Vc3 and fed it forward. Then the machine loops to state S0 if there is a variation of amplitude level from the reference value at output of VGA3. The state diagram for generating the control voltage Vc is shown in Figure 3.30 below.

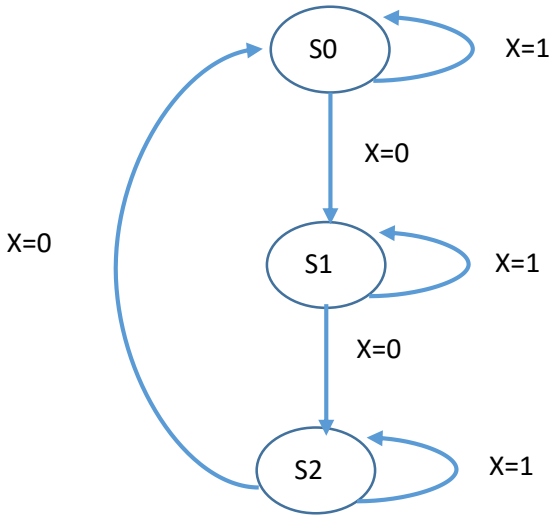


Figure 3.30: State diagram for generating control voltage, V_c.

An algorithm is best expressed using Algorithmic Sate Machine (ASM) as shown in Figure 3.31 below.

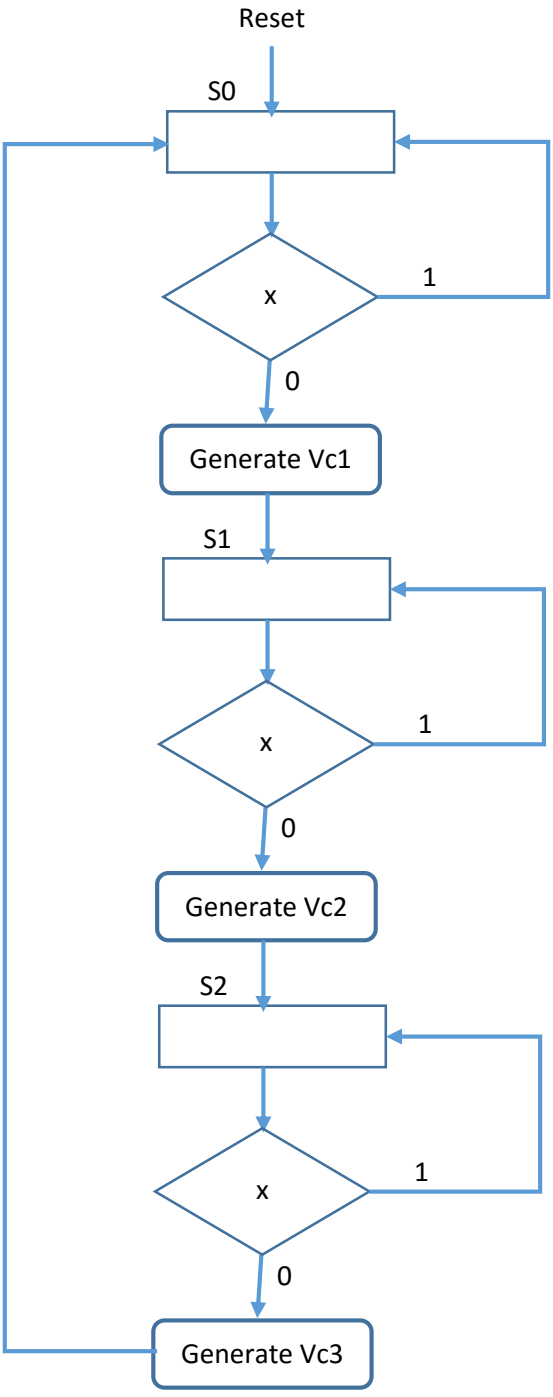


FIGURE 3.31: ASM TO GENERATE CONTROL VOLTAGE V_c FOR EACH VGA STAGE

3.5.3 ARITHMETIC DIVISION IN FPGA

Integer or finite length fractional numbers can be multiplied exactly, whenever sufficient length is allowed for the result. Division doesn't share this feature. As a matter of fact, division generally does not provide a finite length result. The accuracy must be defined beforehand by setting the unit in the least significant position (**ulp**) of the result. The number of algorithmic cycles will therefore depend on the desired accuracy, not on the operand length [24].

The following theorem and algorithm of division is based on reference [24].

Let X and Y be two natural numbers with Y > 0. Define Q and R, respectively, as the quotient and the remainder of the division of X by Y, with an accuracy of p fractional base-B digits:

$B^p.X = Q.Y + R$; where Q and R are natural numbers, and R < Y. The basic algorithm applies to operands X and Y such that X < Y.

In the general case, to ensure that X < Y, a previous alignment step is necessary. Assume that X is an m-digit base-B number, that is, $X < B^m$; then Substitute Y by $Y' = B^m.Y$, so that $Y' \geq B^m.1 > X$; compute the quotient Q and the remainder R' of the division of X by Y', with an accuracy of p+m fractional base-B digits, that is, $B^{p+m}.X = Q.Y' + R'$, with R' < Y' so that $B^p.X = Q.Y + R$, with $R = R'/B^m < Y$.

Since FPGAs work in binary number format, the base-2 division algorithm of reference [24] described below is used in the VHDL code of this thesis work.

For two binary numbers a and b with a<b, the division algorithm of a by b (a/b) is,

```
z:=2*a-b;
r:=a;
for (i=m+p-1 downto 0) loop
if z<0 then
q(i):='0';
r:=2*r;
else
q(i):='1';
r:=z;
end if;
z:=2*r-b
end loop;
```

A VHDL code that computes and generates the control voltages for each Variable Gain Amplifier (VGA) is developed in Xilinx ISE 14.7 as shown in Appendix. The complete system design (analog + digital designs) of the FPGA based AGC circuit as implemented in Matlab-2013 Simulink configured with Xilinx ISE 14.7 system generator is shown in Figure 3.32 below.

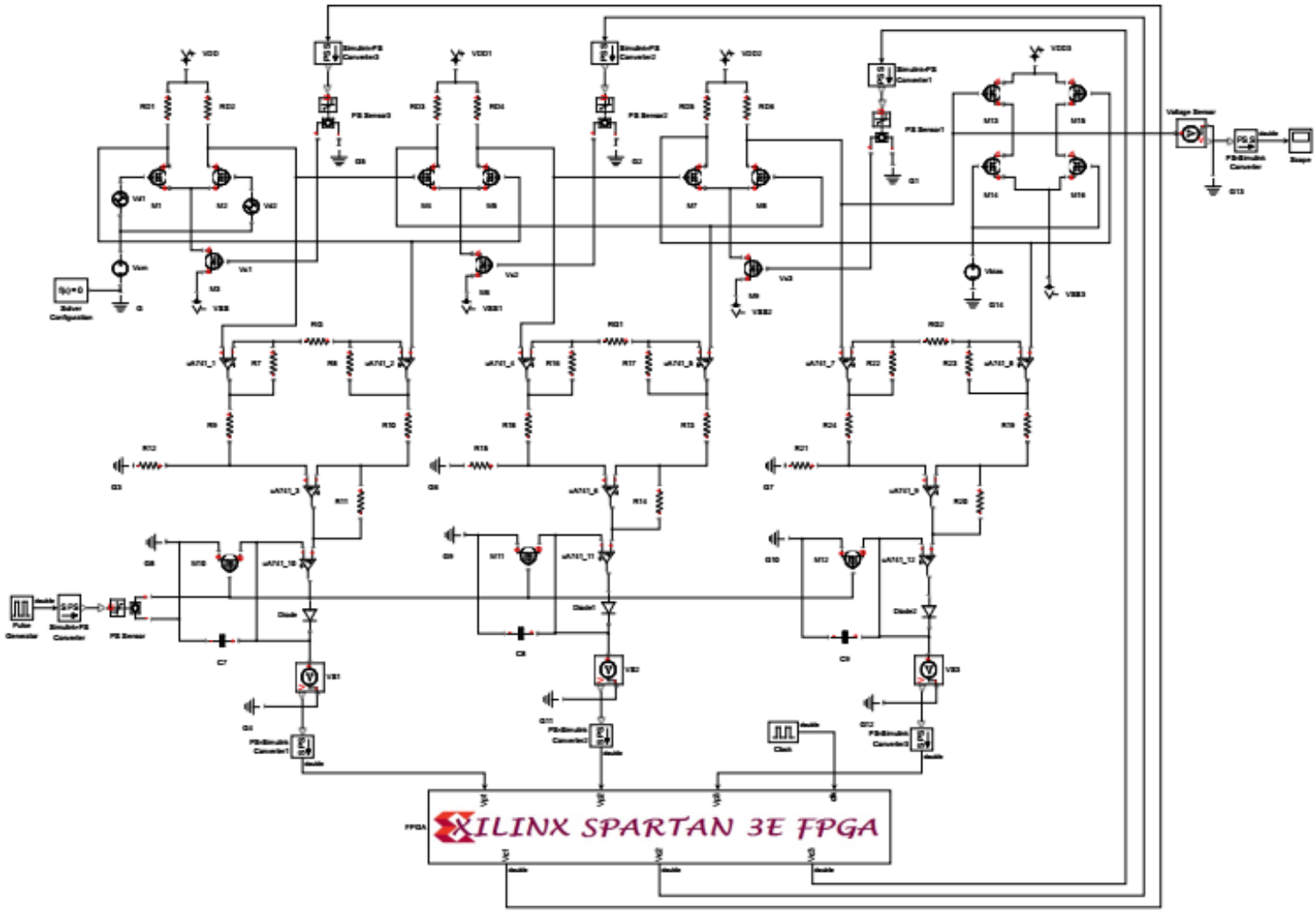


FIGURE 3.32: THE COMPLETE FPGA BASED AGC DESIGN

The Xilinx Spartan 3E shown in Figure 3.32 above is a subsystem in Matlab Simulink and consists of the components shown in Figure 3.33 below.

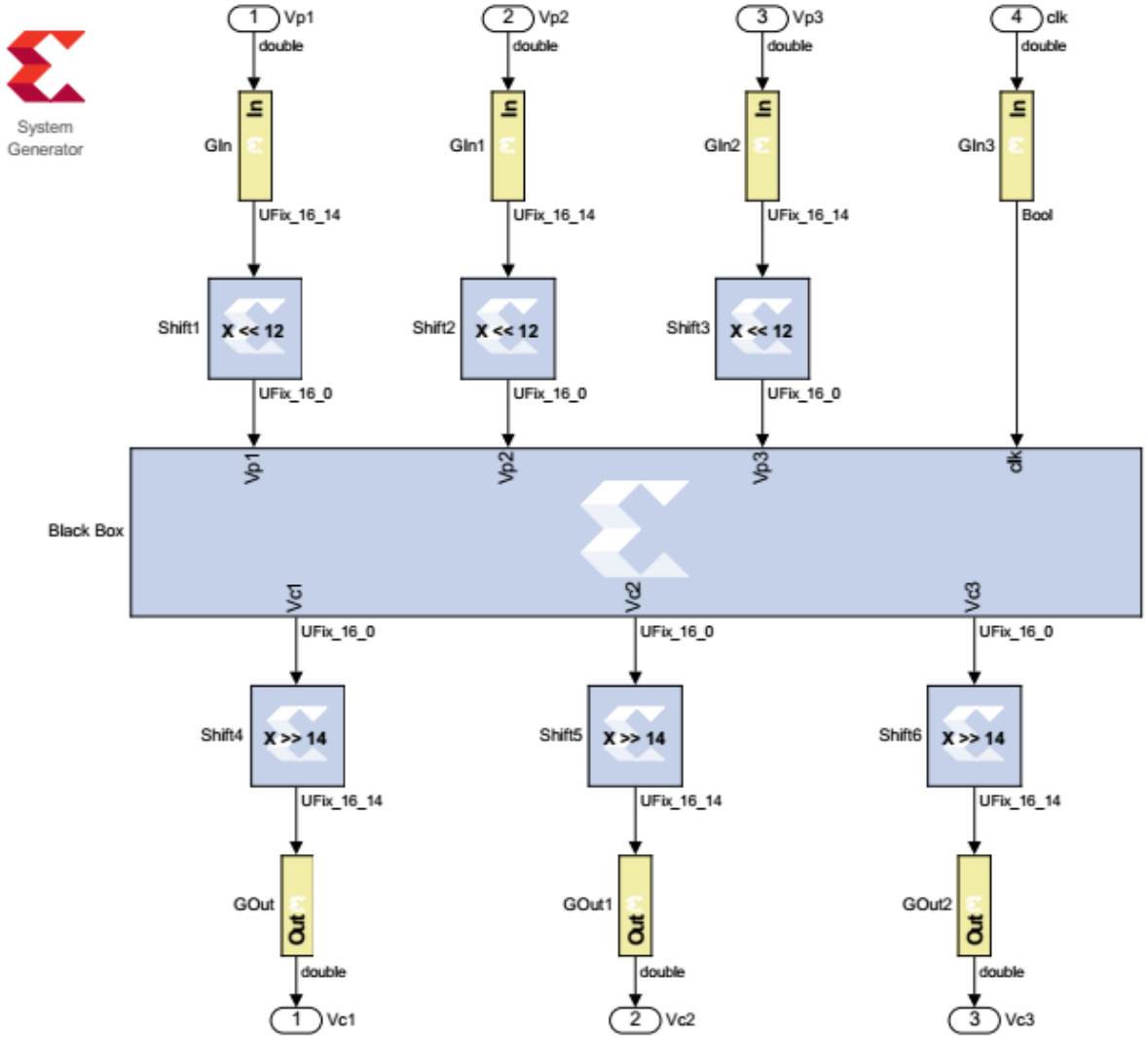


FIGURE 3.33: THE XILINX FPGA SUBSYSTEM AS DESIGNED IN MATLAB SIMULINK

From Figure 3.33 above,

- The blocks labeled with Gin, Gin1, Gin2 and Gin3 are called gateway in blocks and are entry from Matlab to Xilinx System generator. They convert the floating point values in Matlab to fixed point values in FPGA because FPGAs can only compute fixed point values.
- The left shift operations are equivalent to multiplication by 2^{12} and are used to scale up the fixed point values to integer equivalents by 2^{12}. The single-ended output of the differential amplifier was scaled up by $4=2^2$ by the instrumentation amplifier and now it is scaled up by

2^{12} by the left shift operations and then totally scaled up by 2^{14}; therefore, the peak value of the single-ended output at each amplification stage is scaled up by 14 bits before it enters the BlackBox block. The purpose of scaling up by 14 bit is to convert the decimal values to integer values for digital computations inside the FPGA. They will be scaled down by 14 bit to their normal values at the output of the FPGA.

- The BlackBox block accepts the VHDL code shown in Appendix below and generates the control voltages accordingly.
- The right shift operation by 14 bit are equivalent to division by 2^{14} and are used to return back the integer scaled values to their original fixed point values.

CHAPTER 4: MATERIALS AND METHODOLOGY

4.1 MATERIALS USED

- For design simulation, testing and verification of analog circuits, OrCAD PSpice 17.2-2016 is used.
- For VHDL code development, simulation and synthesis of the FPGA, Xilinx ISE Design Suite 14.7 is used.
- For design simulation, testing and verification of the complete system of embedded analog and digital circuits, Matlab-R2013a Simulink configured with Xilinx ISE 14.7 System generator is used. OrCAD PSpice is suitable for analog designs, Xilinx ISE is suitable for digital designs and Matlab Simulink has the capability to simulate integrated analog and digital designs. Accordingly, this thesis work used Matlab Simulink to simulate, test and verify the complete FPGA based AGC circuit.

The following discrete components are also used in the design:

- For VGA design:
  - MOSFETs IRF840, Resistors and Capacitors
- For design of instrumentation amplifier:
  - OPAMPs uA741, and Resistors
- For design of peak detector circuit:
  - MOSFETs IRF840, OPAMPs uA741, and Capacitors
- FPGA Spartan 3E built in Nexys2 Digilent board on which the developed VHDL code is loaded and configured.
- M21-5000 digital training system

4.2 METHODOLOGY

1. Simulation and testing for the analog part of the design is done for 3 amplification stages in OrCAD PSpice 17.2-2016
2. A VHDL code is developed in Xilinx ISE 14.7 according to the algorithm in Figure 3.29 to generate control voltages Vc1, Vc2 and Vc3 based on the sensed peak output values of each stage by the peak detector circuit. Vc1 is fed back to VGA1; Vc2 and Vc3 are fed forward to VGA2 and VGA3 respectively for gain adjustment. A reference differential input voltage of

Vinref=25 mVpeak and an initial control voltage value of Vc = 1.2V are selected for each stage for testing purposes.

3. Finally, simulation and testing of the complete FPGA based AGC system is done in Matlab - R2013a Simulink. Tests are conducted for a differential ac input voltage of Vin=15 mVpeak and 20 mVpeak below the chosen reference input voltage as well as Vin=30 mVpeak and 35 mVpeak above the chosen reference.
4. Hardware implementation as shown in Figure 4.1 below is also implemented and analysis of the output signal amplitude for the selected test values in step-3 above is conducted.

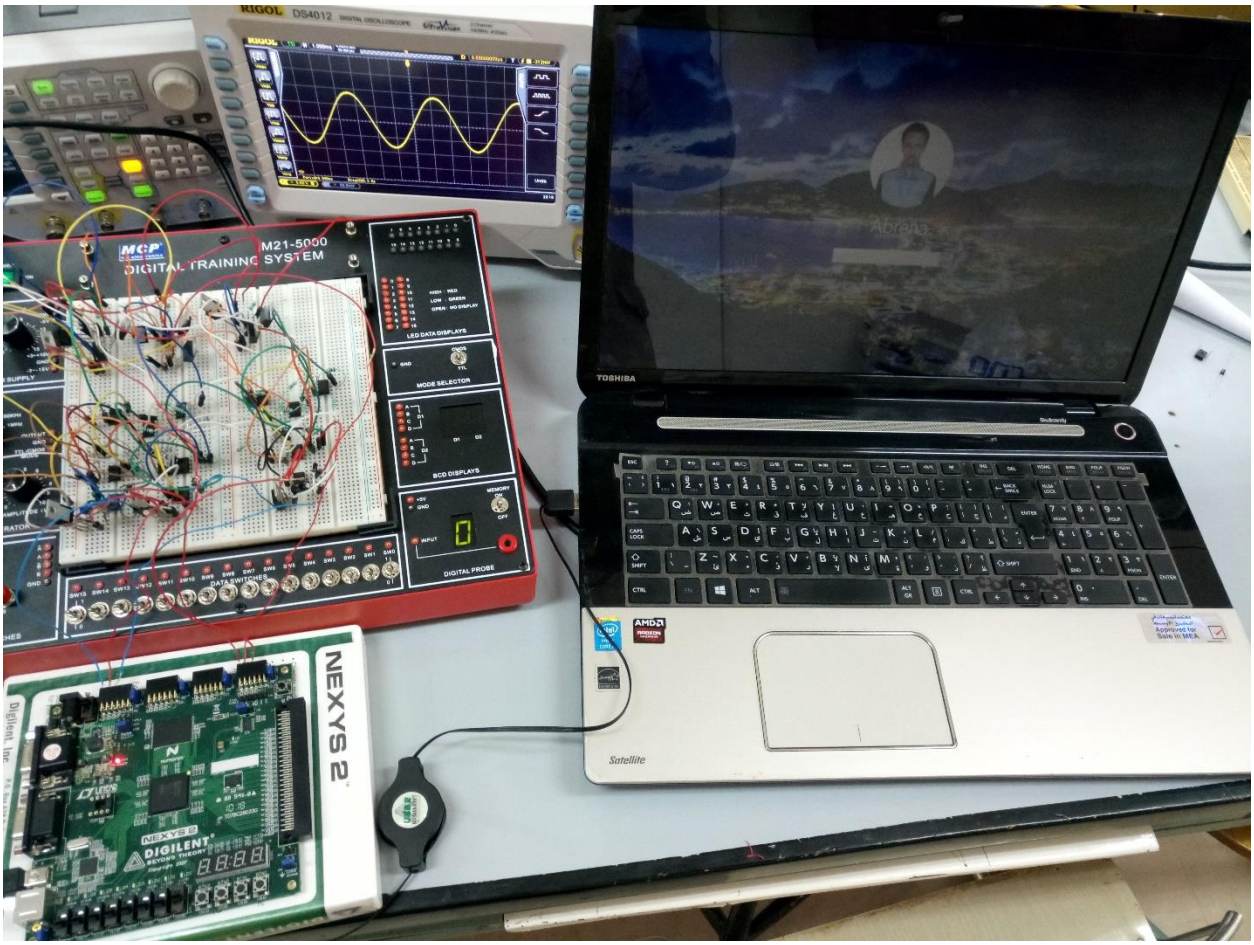


FIGURE 4.1: HARDWARE IMPLEMENTATION OF THE DESIGNED FPGA BASED AGC

CHAPTER 5: RESULTS AND DISCUSSIONS

5.1 FPGA SIMULATION RESULTS

The Spartan 3E FPGA is loaded and configured with the VHDL code shown in Appendix. Development, testing and verification of the code is done in Xilinx ISE 14.7 design suite. Desired ac differential input, Vinref is set to Vinref=25 mVpeak. The initial values of all the three control voltages Vc is set to Vc=1.2 V for testing purposes. From Table 3.2 above for control voltage Vc=1.2, the voltage gain Av=1.5143. The desired output at the last/third stage is, Vref3=Vinref*Av*Av*Av=25 mVpeak* 1.5143^3=86.8 mVpeak=0.0868Vpeak.

Simulation tests are conducted when the ac input fluctuates below and above Vinref. Vin = 20 mVpeak and Vin=30 mVpeak are selected for testing purposes. Let Vp1, Vp2 and Vp3 be the detected peak output voltage values at stage-1, stage-2 and stage-3 respectively.

- When Vin=20 mVpeak:

  Vp1= 20 mV*Av= 20 mV* 1.5143= 30.286 mVpeak=0.030286 Vpeak,

  Vp2=20 mV*Av*Av=20 mV*1.5143^2=45.862 mVpeak=0.045862 Vpeak,

  Vp3=20 mV * Av* Av* Av=20 mV * 1.5143^3= 69.449 mVpeak=0.069449 Vpeak,

  The above peak values are scaled up by 14-bit precision to equivalent integer values for digital computations in FPGA. Let Vp1_scaled, Vp2_scaled and Vp3_scaled are the scaled values of Vp1, Vp2 and Vp3.

  Vp1_scaled=0.030286 *2^{14}= 496= $(111110000)_2$

  Vp2_scaled=0.045862 *2^{14}= 751= $(1011101111)_2$

  Vp3_scaled=0.069449 *2^{14}=1,137= $(10001110001)_2$

  Simulating the FPGA with the above values, the control voltage Vc results shown in Figure 5.1 below are found.

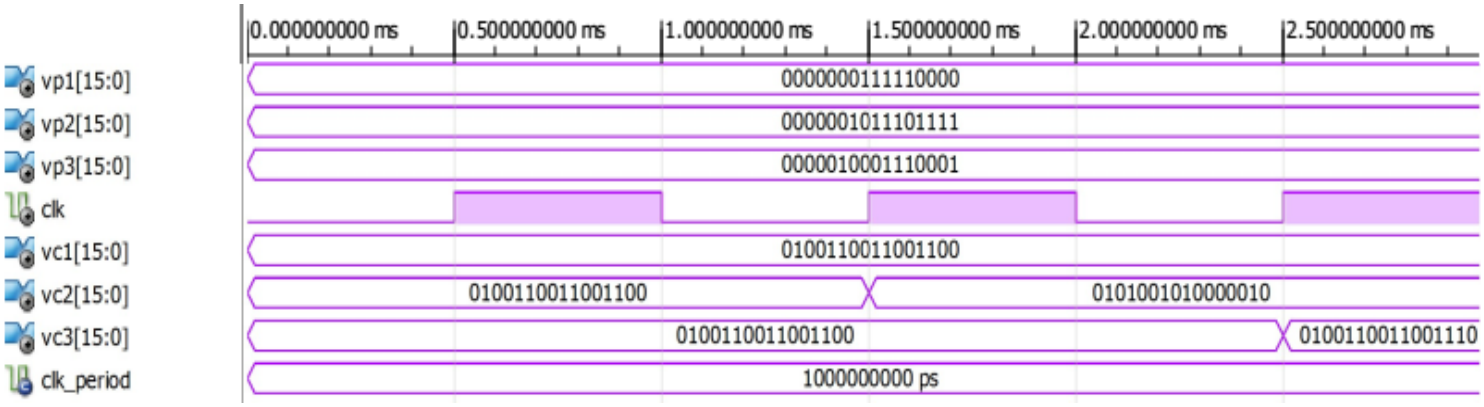


FIGURE 5.1: Xilinx ISE Simulation result for V_{in}=20 mVpeak

As can be seen from diagram above Vc1= $(0100110011001100)_2$= 19660

Vc2= $(0101001010000010)_2$=21122, and

Vc3= $(0100110011001110)_2$= 19662 are generated at last clock cycles.

When these control voltage values are scaled down by 14 bit to their normal decimal values,

Vc1=19,660/2^{14}=1.2V

Vc2=21,122/2^{14}=1.29 V

Vc3= 19662/2^{14}=1.2 V

The validation of the generated control voltages is checked in the integrated analog and digital simulation in Matlab Simulink as discussed in next sections.

- When Vin=30 mVpeak:

  Vp1= 30 mV*Av= 30 mV* 1.5143= 45.429 mVpeak=0.045429 Vpeak,

  Vp2=30 mV*Av*Av=30 mV*1.5143^2= 68.793 mVpeak=0.068793 Vpeak,

  Vp3=30 mV * Av* Av* Av=30 mV * 1.5143^3= 104.173 mVpeak=0.104173 Vpeak,

  The above peak values are scaled up by 14-bit precision to equivalent integer values for digital computations in FPGA. Let Vp1_scaled, Vp2_scaled and Vp3_scaled are the scaled values of Vp1, Vp2 and Vp3.

  Vp1_scaled= 0.045429 *2^{14}= 744= $(1011101000)_2$

  Vp2_scaled=0.068793 *2^{14}= 1127= $(10001100111)_2$

  Vp3_scaled=0.104173 *2^{14}=1706= $(11010101010)_2$

  Simulating the FPGA with the above values, the control voltage Vc results shown in Figure 5.2 below are found.

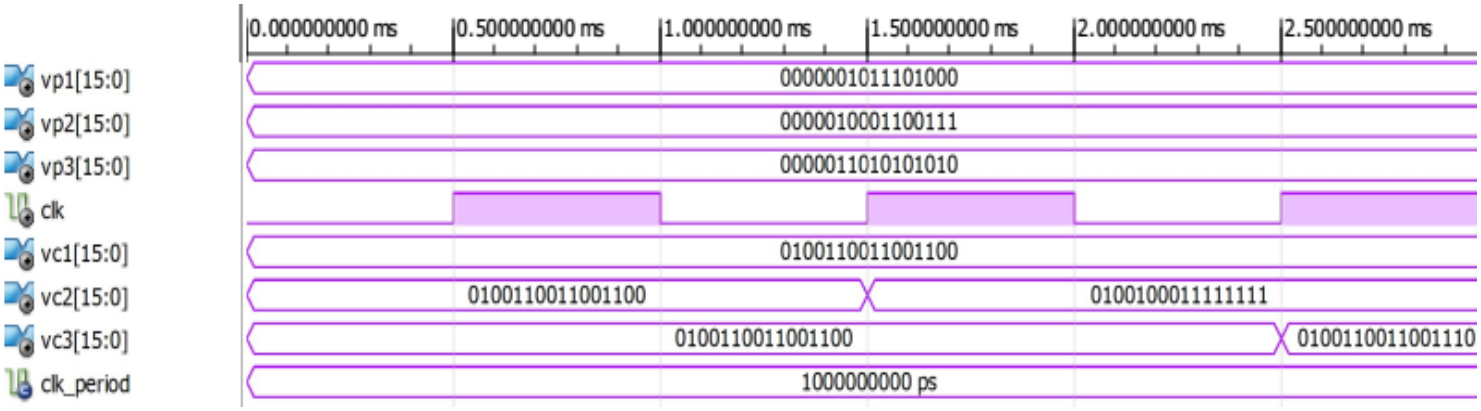


FIGURE 5.2: Xilinx ISE Simulation Result when Vin=30 mVpeak

As can be seen from diagram above Vc1= $(0100110011001100)_2$= 19660

Vc2= $(0100100011111111)_2$=18687, and

Vc3= $(0100110011001110)_2$= 19662 are generated at last clock cycles.

When these control voltage values are scaled down by 14 bit to their normal decimal values,

Vc1=$19660/2^{14}$=1.2 V

Vc2=$18{,}687/2^{14}$=1.14 V

Vc3= $19662/2^{14}$=1.2 V

Similarly, the validation of the generated control voltages is checked in the integrated analog and digital simulation in Matlab Simulink as discussed in next sections.

5.2 MATLAB SIMULINK SIMULATION AND HARDWARE IMPLEMENTATION RESULTS

Software simulation and hardware implementation tests are conducted for the designed AGC circuit and the following results shown in Table 5.1 are obtained for the given inputs. As can be observed from the table, a constant amplitude output signal is achieved even though the input signal to the AGC circuit fluctuates below and above the reference/desired amplitude level.

TABLE 5.1: SIMULATION AND HARDWARE IMPLEMENTATION RESULTS

Input signal amplitude, Vip (mV)	Calculated output signal amplitude without gain adjust circuit (FPGA) (mV)	Desired output signal amplitude, Vop_ref (mV)	Simulated output signal amplitude, Vop with gain adjust circuit (FPGA) (mV)	Hardware tested output signal amplitude, Vop with gain adjust circuit (FPGA) (mV)	Remark
15	52.07	86.8	87	85	
20	69.45	86.8	87	85	
25	86.8	86.8	87	85	NB: this is the reference input
30	104.17	86.8	87	85	
35	121.54	86.8	87	85	

The output Vopeak is the output in the last stage of the AGC circuit. Sample simulation figures for Vin=20 mVpeak and Vin=30 mVpeak are shown below.

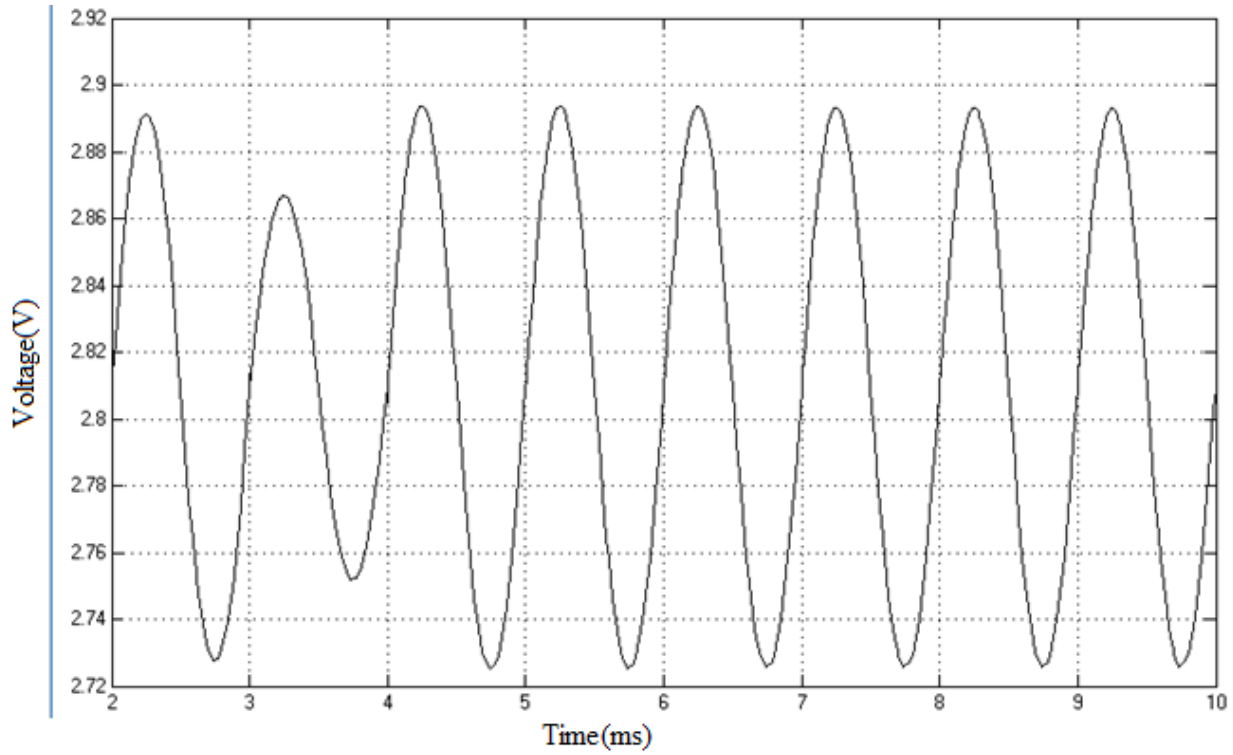


FIGURE 5.3: Matlab Simulink Simulation Result for Vin =20 mVpeak

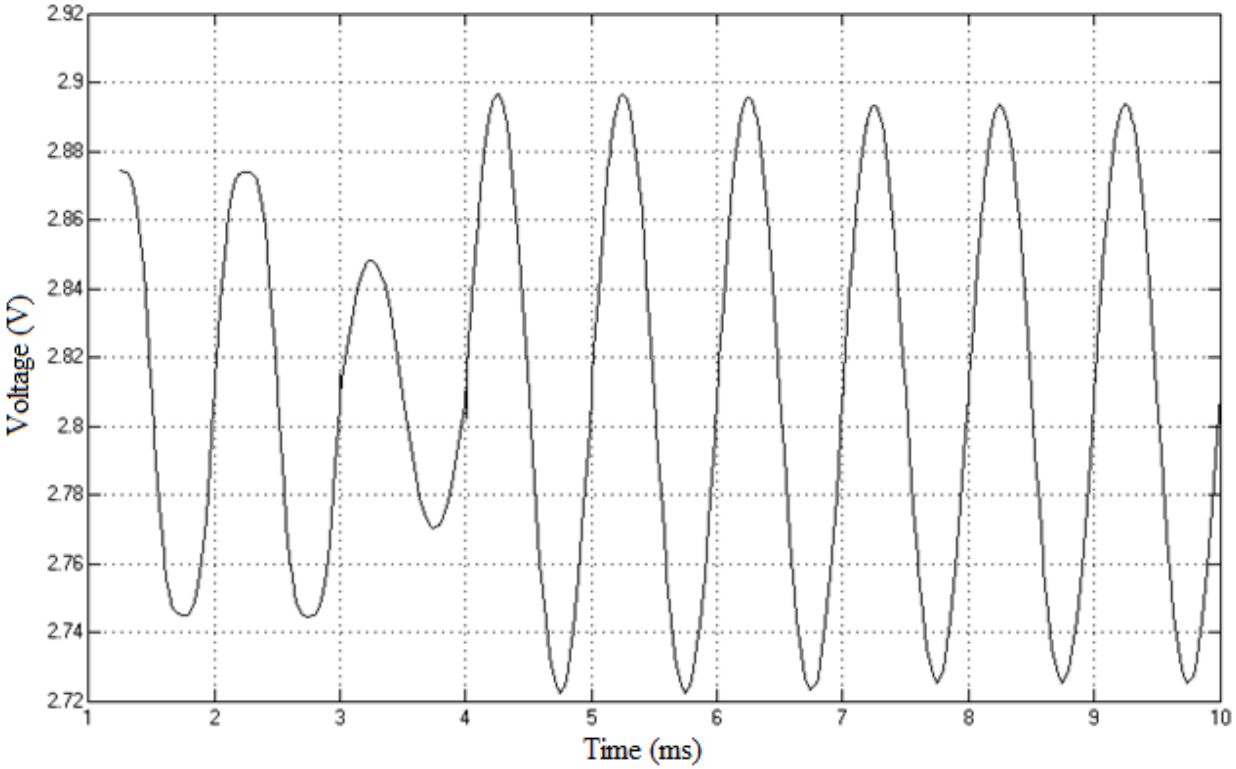


FIGURE 5.4: Matlab Simulink Simulation Result for Vin =30 mVpeak

As can be seen from figures 5.3 and 5.4 above, the output is settled to the desired value after 4 ms. Besides, the peak amplitude value of the two signals is the same since gain adjustment is done by the FPGA for constant amplitude output signal.

CHAPTER 6: CONCLUSION AND RECOMMENDATION

6.1 CONCLUSION

Though the input may vary below and above the reference level, the designed FPGA based AGC circuit take control of the amplitude variation and delivers a constant amplitude output signal. Using FPGAs to generate the control voltages has the advantages of flexibility over ASICs and better performance speed, less area consumption and high power efficiency over microcontrollers. Feedback and feedforward loops of gain adjustment are used in this work to utilize the strong sides of both closed loop mechanisms. Using multistage enhances the dynamic range of the overall gain. In case of this work, the range is enhanced from 21 dB for single stage to 63 dB for three stages. Though the input varies below and above the input reference value which is 25 mvPeak, a constant amplitude signal of 87 mVpeak is achieved for all inputs. The designed FPGA based AGC circuit can be applied in communication systems where the faded transmitted signal is adjusted to desired amplitude level at receiving end, in scanning devices for consistent image qualities, and in disc drive read channels for consistent sound output, etc.

6.2 RECOMMENDATION FOR FUTURE WORKS

The following points are recommended for future works around this thesis,

- Design the Variable Gain Amplifier (VGA) fully from MOSFETs for better controllability, less power dissipation and better output voltage swing.
- Apply the AGC circuit for specific real world applications that are not discussed in this paper.

REFERENCES

[1] H.-C. CHOW and I.-H. WANG, "High Performance Automatic Gain Control Circuit for communication applications," Taiwan.

[2] P. J. Alegre, S. Celma and B. Lopez, "AGC Fundamentals," in *Automatic Gain Control Techniques and Architectures for RF Receivers*, Springer Science+Business Media, LLC, 2011, pp. 13-28.

[3] J. P. Alegre, B. Calvo and S. Celma, "A Low-Voltage High-Performance CMOS Feedforward AGC Circuit for Wideband Wireless Receivers," *ETRI Journa,* vol. 30, no. 5, October 2008.

[4] X. Cheng, G. Xie, Z. Zhang and Y. Yang, "Fast-Settling Feedforward Automatic Gain Control Based on a New gain Control Approach," *IEEE,* vol. 61, no. 9, September 2014.

[5] I. Rosu, "QSL.NET," [Online]. Available: http://www.qsl.net/va3iul. [Accessed October 2015].

[6] J. R. Giebel, "Digital Automatic Gain Cintrol for Multi-Stage Amplification Circuits". USA/new York Patent 6,073,848, 13 June 2000.

[7] S. Hauck and A. DeHon, Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation, Burlington: Morgan KaufMann Publications, 2008.

[8] L. Hang, "Design of Variable Gain Amplifier in CMOS Technology," 2015.

[9] E. Miletiü, M. Krstiü, M. Piz and M. Methfessel, "Digital Automatic Gain Control Integrated on WLAN platform," *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering,* vol. 2, no. 5, 2008.

[10] J. P. Alegre, B. Calvo and S. Celma, "A Fast Compact CMOS Feedforward Automatic Gain Control Circuit," *IEEE,* 2008.

[11] V. P. G. Jiménez, M. J. F.-G. García, F. J. G. Serrano and A. G. Armada, "Design and Implementation of Synchronization and AGC for OFDM-based WLAN receivers," *IEEE.*

[12] K. Sobaihi, A. Hammoudeh and D. Scammell, "Automatic Gain Control on FPGA for SoftwareDefined Radios," *IEEE,* 2012.

[13] B. Bloessl, C. Sommer and F. Dressler, "Power Matters: Automatic Gain Control for a Software Defined Radio IEEE 802.11a/g/p Receiver," *IEEE,* 2015.

[14] J. Gang, Z. Yiqi, Y. Yue and C. Miao, "A digitally controlled AGC loop circuitry for GNSS receiver chip with a binary Weighted Accurate dB-linear PGA," *Journal of Semiconductors,* vol. 36, no. 3, March 2015.

[15] X. Chen and M. Du, "Automatic Gain Control for Multi-channel Electrochemical Detection System Combining FPGA and FPAA," in *International Conference on BioMedical Engineering and Informatics (BMEI)*, 2012.

[16] E. Salas and P. Alvarado, "lementation of an automatic gain control for audio signals in an application for environmental protection," in *Conference on Technologies for Sustainable Development*, 2011.

[17] B. Razavi, Design of Analog CMOS Intehrated Circuits, Newyork: McGraw-Hill, 2001.

[18] B. Razavi, Fundamentals of Microelectronics, 2006.

[19] T. L. Floyd, Electronic Devices, 9th ed., New Jersey: Prentice Hall, 2012.

[20] P. E. Allen and D. R. Holberg, CMOS Analog Circuit Design, 2nd ed., New York: Oxford University Press, 2002.

[21] M. J. Roberts, Signals and Systems: Analysis Using Transform Methods and MATLAB, 2nd ed., New York: MC Graw-Hill, 2012.

[22] S. Hauck and A. DeHon, Eds., Reconfigurable Computing: The Theory and Practice of FPGA-Based Computing, Burlington: Elsevier Inc, 2008.

[23] S. Brown and Z. Vranesic, Fundamentals of Digital Logic with VHDL Design, 2nd ed., New York: McGraw-Hill, 2005.

[24] J.-P. Deschamps, G. J. A. Bioul and G. D. Sutter, Synthesis of Arithmetic Circuits: FPGA, ASIC and Embeded Systems, New Jersey: John Wiley & Sons, Inc., 2006.

APPENDIX: VHDL CODE TO GENERATE CONTROL VOLTAGES

```
----------------------------------------------------------------------------------
-- MSc thesis FPGA implementation
--Institute: AAU
--Title: FPGA Based Automatic Gain Control Circuit Design for Multistage Amplification
-- Student: Abreha Teklu
-- Advisor: Prof. Mohammed Abdo
-- Module Name:    generateVc - agc_arc
-- Target Devices: Spartan 3E
----------------------------------------------------------------------------------
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.NUMERIC_STD.ALL;
ENTITY generateVc IS
   PORT (Vp1 : IN  STD_LOGIC_VECTOR (15 DOWNTO 0);--Vp scaled by 14 bit
            Vp2 : IN  STD_LOGIC_VECTOR (15 DOWNTO 0);
            Vp3 : IN  STD_LOGIC_VECTOR (15 DOWNTO 0);
            clk : IN STD_LOGIC;
            Vc1 : OUT  STD_LOGIC_VECTOR (15 DOWNTO 0):="0100110011001100";
                                                    --initialize to 1.2 scaled by 14-bit precision
            Vc2 : OUT  STD_LOGIC_VECTOR (15 DOWNTO 0):="0100110011001100";
            Vc3 : OUT  STD_LOGIC_VECTOR (15 DOWNTO 0):="0100110011001100");
END generateVc;
ARCHITECTURE agc_arc OF generateVc IS
---All values are scaled by 14 bit precision----------
---the following constants are calculated assuming differential ac input reference
--Vinref=25 mVpeak;
CONSTANT Vc0: UNSIGNED(31 DOWNTO 0):= TO_UNSIGNED(INTEGER(1.2*16384.0),32);
CONSTANT k1: UNSIGNED(17 DOWNTO 0) :=TO_UNSIGNED(INTEGER(4.2524*16384.0),18);
CONSTANT k2: UNSIGNED(15 DOWNTO 0) :=TO_UNSIGNED(INTEGER(0.8439*16384.0),16);
CONSTANT k3: UNSIGNED(15 DOWNTO 0) :=TO_UNSIGNED(INTEGER(0.2352*16384.0),16);
CONSTANT Av0: UNSIGNED(31 DOWNTO 0) := TO_UNSIGNED(INTEGER(1.5143*16384.0),32);
CONSTANT Vref1: UNSIGNED(15 DOWNTO 0):= TO_UNSIGNED(INTEGER(0.0379*16384.0),16);
CONSTANT Vref2: UNSIGNED(15 DOWNTO 0) :=TO_UNSIGNED(INTEGER(0.0573*16384.0),16);
```

```
CONSTANT Vref3: UNSIGNED(15 DOWNTO 0):=TO_UNSIGNED(INTEGER(0.0868*16384.0),16);
SIGNAL Vp1_unsigned      : UNSIGNED(15 DOWNTO 0);
SIGNAL Vp2_unsigned      : UNSIGNED(15 DOWNTO 0);
SIGNAL Vp3_unsigned      : UNSIGNED(15 DOWNTO 0);
SIGNAL Av1               : UNSIGNED(31 DOWNTO 0)      :=Av0;
SIGNAL Av2               : UNSIGNED(31 DOWNTO 0)      :=Av0;
SIGNAL Av3               : UNSIGNED(31 DOWNTO 0)      :=Av0;
CONSTANT Vcl : UNSIGNED(31 DOWNTO 0) :=TO_UNSIGNED(INTEGER(0.9*16384.0),32);
CONSTANT Vch : UNSIGNED(31 DOWNTO 0) :=TO_UNSIGNED(INTEGER(1.5*16384.0),32);
SIGNAL Vc1_flag, Vc2_flag, Vc3_flag: BIT:='0';--flags to handle delta delays
SIGNAL init: BIT:='1';
BEGIN
Vp1_unsigned <= UNSIGNED(Vp1);
Vp2_unsigned <= UNSIGNED(Vp2);
Vp3_unsigned <= UNSIGNED(Vp3);
-----------------------------------process----------------------------------
procVc : PROCESS (clk)
VARIABLE Vc1_tmp          : UNSIGNED(31 DOWNTO 0) :=Vc0;
VARIABLE Vc2_tmp          : UNSIGNED(31 DOWNTO 0) :=Vc0;
VARIABLE Vc3_tmp          : UNSIGNED(31 DOWNTO 0) :=Vc0;
VARIABLE Av1_adjusted : UNSIGNED(31 DOWNTO 0);
VARIABLE Av2_adjusted : UNSIGNED(31 DOWNTO 0);
VARIABLE Av3_adjusted : UNSIGNED(31 DOWNTO 0);
VARIABLE Q1: UNSIGNED(29 DOWNTO 0);--quotient of Vref1/Vp1 during feedback
VARIABLE Q2: UNSIGNED(29 DOWNTO 0);--quotient of Vref2/Vp2 during feedforward
VARIABLE Q3: UNSIGNED(29 DOWNTO 0);--quotient of Vref3/Vp3 during feedforward
VARIABLE Vp1_scaled: INTEGER;
VARIABLE Vp2_scaled: INTEGER;
VARIABLE Vp3_scaled: INTEGER;
VARIABLE i,r,z: INTEGER;
BEGIN
IF(clk='1' AND (init='1' OR Vc3_flag='1')) THEN
-----generate Vc1 using feedback loop-----
IF(Vp3_unsigned = Vref3) THEN
```

```
Vp1_scaled :=TO_INTEGER(Vp1_unsigned)*65536; --multiply by 2^16 to insure
                                                --Vp>Vref for division algorithm
z:=2*TO_INTEGER(Vref1)-Vp1_scaled;
r:=TO_INTEGER(Vref1);
Q1:="00000000000000000000000000000000";--initialize to 0
--base 2 division to find Vref1/Vp1 with 16+14 iterations
FOR i IN 29 DOWNTO 0 LOOP
IF z<0 THEN
Q1(i):='0';
r:=2*r;
ELSE
Q1(i):='1';
r:=z;
END IF;
z:=2*r-Vp1_scaled;
END LOOP;
----Actual Q1:= Q1/2^14 but left for integer scaling;
Av1_adjusted := RESIZE((Av1* Q1)/16384,32);-- holds adjusted gain value
Vc1_tmp :=RESIZE(((k3*Av1_adjusted)/16384+k2),32);
IF(Vc1_tmp<Vcl) THEN-- if Vc1 is below minimum value of control voltage
  Vc1_tmp :=Vcl;
ELSIF(Vc1_tmp>Vch) THEN-- if Vc1 is above maximum value of control voltage
  Vc1_tmp :=Vch;
END IF;
Vc1 <=STD_LOGIC_VECTOR(RESIZE(Vc1_tmp,16));--resize to 16 bit
Av1<=RESIZE((k1*(Vc1_tmp-k2))/16384,32); -- adjust VGA1
END IF;
Vc1_flag<='1';
init<='0';
Vc3_flag<='0';
END IF;
IF(clk='1' AND Vc1_flag='1') THEN
-----generate Vc2 using feedforward loop-----
IF(Vp1_unsigned /= Vref1) THEN
```

```
Vp2_scaled :=TO_INTEGER(Vp2_unsigned)*65536; --multiply by 2^16 to insure
                                                --Vp>Vref for division algorithm
z:=2*TO_INTEGER(Vref2)-Vp2_scaled;
r:=TO_INTEGER(Vref2);
Q2:="00000000000000000000000000000000";--initialize to 0
FOR i IN 29 DOWNTO 0 LOOP --14+10 bit precision
IF z<0 THEN
Q2(i):='0';
r:=2*r;
ELSE
Q2(i):='1';
r:=z;
END IF;
z:=2*r-Vp2_scaled;
END LOOP;
--Actual Q2:=Q2/2^14 but left for integer scaling;
Av2_adjusted:= RESIZE((Av2*Q2)/16384,32);
Vc2_tmp :=RESIZE(((k3*Av2_adjusted)/16384+k2),32);
IF(Vc2_tmp<Vcl) THEN-- if Vc1 is below minimum value of control voltage
  Vc2_tmp :=Vcl;
ELSIF(Vc2_tmp>Vch) THEN -- if Vc1 is above maximum value of control voltage
   Vc2_tmp :=Vch;
END IF;
Vc2 <=STD_LOGIC_VECTOR(RESIZE(Vc2_tmp,16));
Av2<=RESIZE((k1*(Vc2_tmp-k2))/16384,32); -- adjust VGA2
END IF;
Vc2_flag<='1';
Vc1_flag<='0';
END IF;
IF(clk='1' AND Vc2_flag='1') THEN
-----generate Vc3 using feedforward loop-----
IF(Vp2_unsigned /=Vref2) THEN
Vp3_scaled := TO_INTEGER(Vp3_unsigned)*65536; --multiply by 2^16 to
                                --insure Vp>Vref for division algorithm
```

```
z:=2*TO_INTEGER(Vref3)-Vp3_scaled;
r:=TO_INTEGER(Vref3);
Q3:="00000000000000000000000000000000";--initialize to 0
FOR i IN 29 DOWNTO 0 LOOP --14+10 bit precision
IF z<0 THEN
Q3(i):='0';
r:=2*r;
ELSE
Q3(i):='1';
r:=z;
END IF;
z:=2*r-Vp3_scaled;
END LOOP;
--Actual Q3:=Q3/2^14 but left for integer scaling;
Av3_adjusted:=RESIZE((Av3*Q3)/16384,32);
Vc3_tmp :=RESIZE(((k3*Av3)/16384+k2),32);
   IF(Vc3_tmp<Vcl) THEN-- if Vc1 is below minimum value of control voltage
      Vc3_tmp :=Vcl;
   ELSIF(Vc3_tmp>Vch) THEN -- if Vc1 is above maximum value of control voltage
      Vc3_tmp :=Vch;
   END IF;
Vc3 <=STD_LOGIC_VECTOR(RESIZE(Vc3_tmp,16));
Av2<=RESIZE((k1*(Vc2_tmp-k2))/16384,32); -- adjust VGA3
END IF;
Vc3_flag<='1';
Vc2_flag<='0';
END IF;
END PROCESS procVc;
END ARCHITECTURE agc_arc;
```