Digitally Controlled Average Current Mode Buck Converter

by

Chao Fu

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2011 by the
Graduate Supervisory Committee:

Bertan Bakkaloglu, Chair
Yu Cao
Bert Vermeire

ARIZONA STATE UNIVERSITY

December 2011

ABSTRACT

During the past decade, different kinds of fancy functions are developed in portable electronic devices. This trend triggers the research of how to enhance battery lifetime to meet the requirement of fast growing demand of power in portable devices. DC-DC converter is the connection configuration between the battery and the functional circuitry. A good design of DC-DC converter will maximize the power efficiency and stabilize the power supply of following stages. As the representative of the DC-DC converter, Buck converter, which is a step down DC-DC converter that the output voltage level is smaller than the input voltage level, is the best-fit sample to start with. Digital control for DC-DC converters reduces noise sensitivity and enhances process, voltage and temperature (PVT) tolerance compared with analog control method. Also it will reduce the chip area and cost correspondingly. In battery-friendly perspective, current mode control has its advantage in over-current protection and parallel current sharing, which can form different structures to extend battery lifetime.

In the thesis, the method to implement digitally average current mode control is introduced; including the FPGA based digital controller design flow. Based on the behavioral model of the close loop Buck converter with digital current control, the first FPGA based average current mode controller is burned into board and tested. With the analysis, the design metric of average current mode control is provided in the study. This will be the guideline of the parallel structure of future research.

DEDICATION

To my family

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor Dr. Bertan Bakaloglu, for his continuous support during this period of time. Without his help, I will never finish this work. His trust in my capability inspires me to move forward.

I am grateful to Dr. Yu Cao and Dr. Bert Vermeire for being members on my defense committee. And I really appreciate the efforts they made on reviewing my thesis.

Also, I want to thank Tao Liu for helping me through the whole process, including soldering and board test. With his help, I can integrate the controller on the board.

Last but not least, I am indebted to my family for their unconditional love and support. I'd like to share the important time with all of my friends.

TABLE OF CONTENTS

Page

CHAPTER

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure

Page

CHAPTER 1: INTRODUCTION

1.1. Battery Lifetime

During the past decade, different kinds of fancy functions are developed in portable electronic devices. This trend triggers the research of how to enhance battery lifetime to meet the requirement of fast growing demand of power in portable devices. The figure below is a typical battery discharge characteristic. If the battery is discharging intermittently, the relaxation phenomena can be observed. As can be seen in the figure 1.1, the purple line is the discharge characteristic with relaxation phenomena. When the battery stops discharging, the battery capacity will recover a little bit.

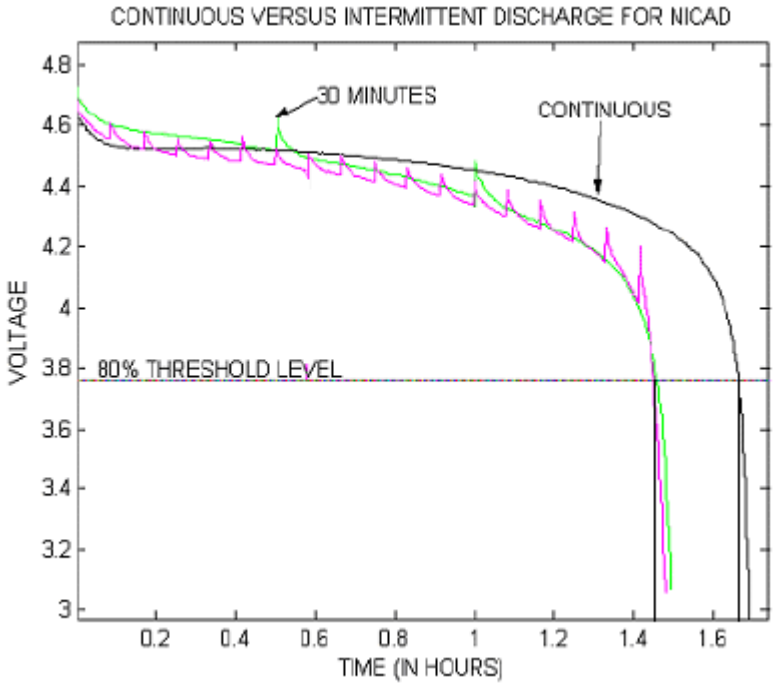

Figure 1.1: Discharge Characteristics of Nickel Cadmium Battery [1]

The best way to make full use of this characteristic is to use multiple batteries paralleled together. In this sense, current mode feedback is the appropriate one for this structure compared with voltage feedback.

1.2. Overview of DC-DC Converter

DC-DC converters are devices which are supplied with power from batteries. It regulates the output voltage level from different input voltage level. It is also called voltage regulators. Since the converter output is usually performed as supply of follow-up circuits, linearity and efficiency are two of important measures in DC-DC converter design. According to the structure differences, basically converters can be divided in two groups.

First type is linear regulator:

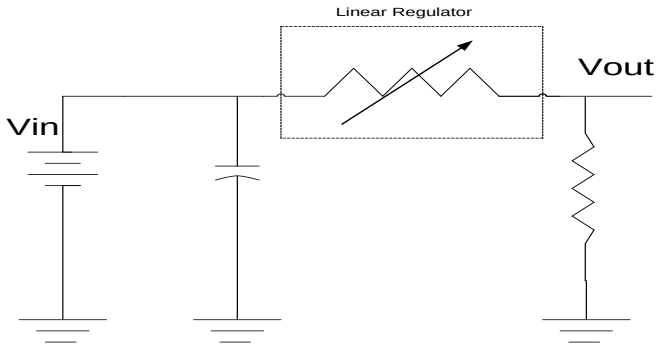

Figure 1.2: Linear regulator

The basic idea of linear regulator is a voltage divider, output will always smaller than the input. The benefit of this structure is very simple and linear while the shortcoming is low efficiency.

$$Vout = Vin * \frac{R_{load}}{R_{load} + R_{regulate}} \qquad (1.1)$$

According to the equation above, the efficiency will suffer a great loss when load is comparable to the regulation resistance. However, when load resistance is much larger than the regulation resistance, linear regulation is a popular in application such as LDO (Low Drop-Out Regulator).

In order to solve the problems in the linear regulator, another type of regulator comes into the picture. That is switch mode regulator, which stores input energy in certain amount of time and then releases the energy to output at different voltage through inductors, capacitors or transformers. The regular efficiency of switch mode regulator is about 75%-98%, which is much higher than that in linear regulator. There is some comparison between the linear regulator and switch mode regulator:

Table 1: Comparison between Linear Regulators with Switch Mode

	Linear Regulator	Switch Mode Regulator
Output Voltage	Always lower than the input with same polarity	Can be higher or lower than input with different polarity
Efficiency Characteristic	Light load: High efficiency; Heavy load: Low efficiency; Generally: much lower than Switch mode	Light load: Low efficiency; Heavy load: High efficiency.
Noise Immunity	Very good	Switching noise; Output has periodical ripples

The switch mode regulator can be further divided into several sub-groups in figure 1.3:

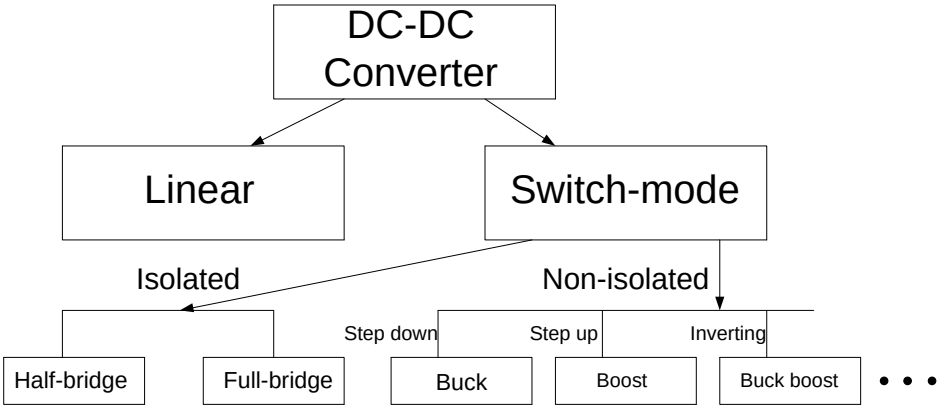

Figure 1.3: Category of DC-DC Converter

Different structure has its special advantages, in this thesis, only buck converter, which is the representative of the switch mode converter families, is discussed and analyzed.

1.3. Buck Converter

The Buck converter structure is shown in figure 1.4:

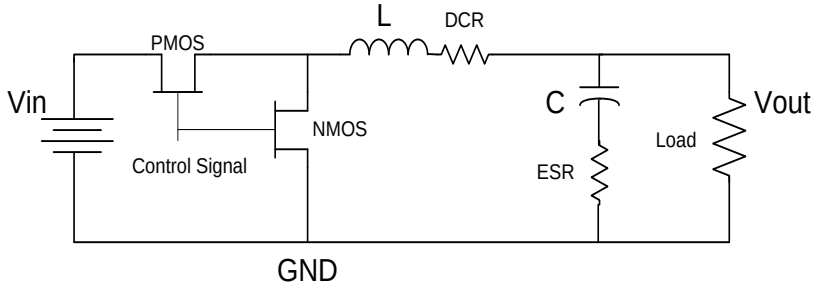

Figure 1.4: Buck Converter Circuitry

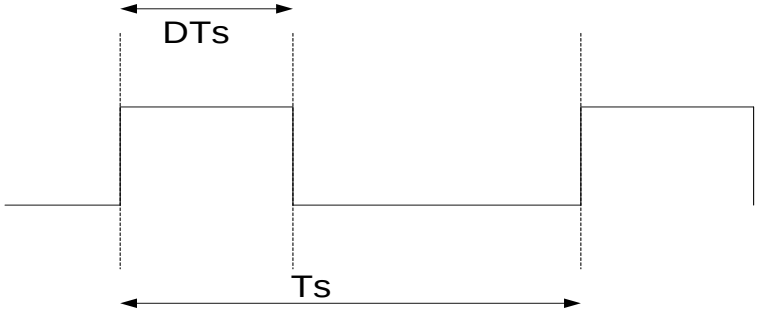

Figure 1.5: Control Signal

The control signal is a pulse with adjusted pulse width in Figure 1.5. PMOS and NMOS transistor cannot conduct at the same time. Define that the PMOS conducting is DTs. During this time, input is charging the capacitor and output. And during the (1-D)*Ts, which means the PMOS is close and NMOS is open, input is cut off and capacitor is charging the output to maintain the load current. This is a full cycle.

The Buck converter is a step down converter. In order to generate an intuitive relationship between input and output, the parasitic resistance like DCR and ESR is ignored in this analysis. Considering the steady state behavior,

$$V_{in} * DT_S = V_o * T_S \tag{1.2}$$

$$V_o = D * V_{in} \tag{1.3}$$

1.4. Control of Buck Converter

The output of DC-DC converter is usually regulated within a specific range in response to the variation of input voltage or load current. Therefore, a feedback control loop is required in DC-DC converter design. The structure is shown in figure 1.6 below:

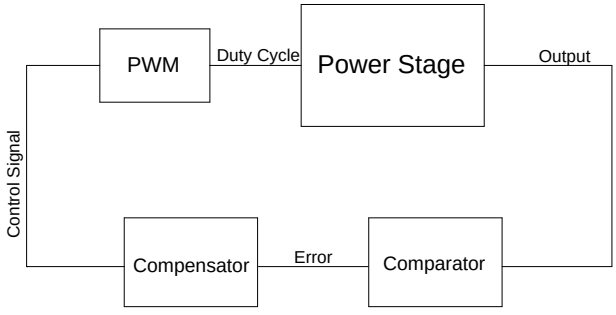

Figure 1.6: Close loop Buck Converter Modules

There are two kinds of disturbance to measure the control loop:

Load regulation: When load current has a jump, the transient response of output voltage;

Line regulation: When input voltage has a jump, the transient response of output voltage.

The better the loop designed, the smaller the output overshoot, the shorter the output settling time.

Considering different sampling method at output node, there are basically two control methods: voltage control and current control.

In voltage mode control, the sampled output voltage is compared with a reference voltage and generates an error signal. Through the compensator, duty cycle will response as the error changes. Suppose that output voltage is higher than the reference, a negative error is generated. This will reduce the duty cycle to close PMOS longer time. As D becomes smaller, output voltage will decrease and error will become smaller. Usually the compensator in voltage control is a PID compensator.

Current mode control includes voltage sampling and current sampling, which is a two-loop control. Voltage sampling is same as voltage mode. Current sampling is to sample the voltage across a sensing resistor. In that way, the inductor ripple current will translate into ramp voltage and being fed back to the compensator. According to different sampling points, the current mode control includes peak current mode (sampling at the connection of PMOS and inductor) and average current control (sampling at the connection of inductor and load)

1.5. Contribution of the Work

This is first average current mode digital controller designed which is burned with actual board.

The average current mode controlled Buck converter is detailed explained. And a highly compatible PID control FPGA model is presented with Simulink. Through the board test, the output regulation with different control schemes is presented. With these results, further research on current sharing and over-current protection will have a valuable reference.

1.6. Thesis Outline and Organization

The organization of the thesis is as follows:

Chapter 2 presents the analog average current mode Buck converter design and comparison with voltage mode and peak current mode control;

Chapter 3 presents analog and corresponding digital Simulink model and simulation results;

Chapter 4 shows the design flow of the PID implementation in FPGA;

Chapter 5 presents the board test results;

Chapter 6 summarizes the thesis.

CHAPTER 2

AVERAGE CURRENT MODE BUCK CONVERTER

2.1. Circuitry Structure

The whole circuit includes basically 3 parts: Power stage, Compensator, PWM generator.

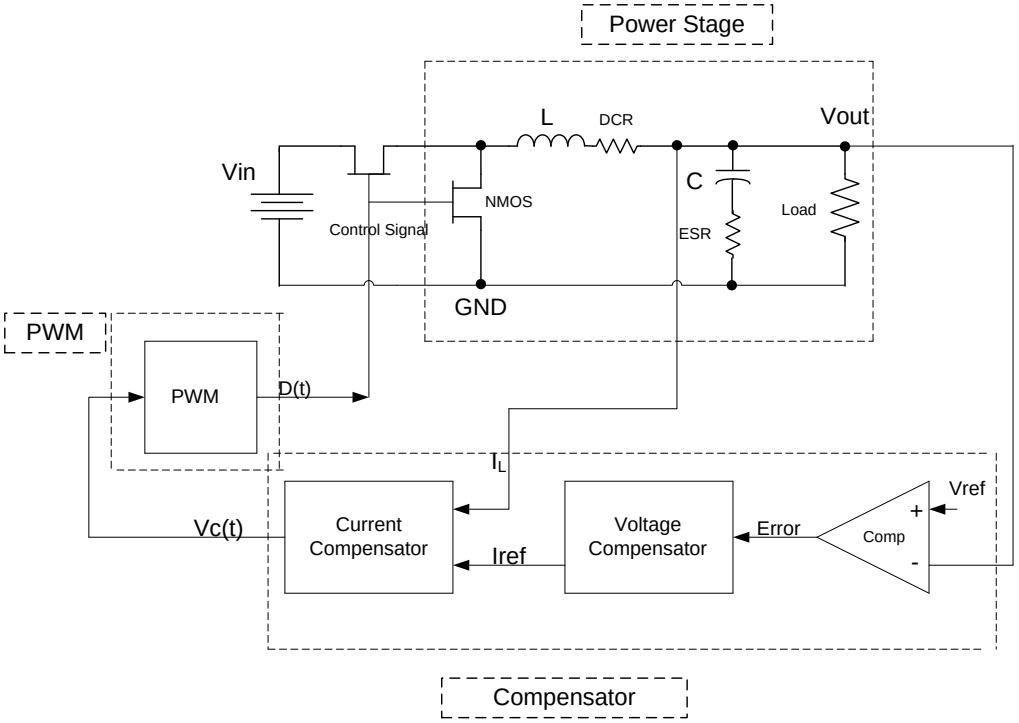

Figure 2.1: Close Loop Buck Circuitry

The outer voltage loop provides current reference of inner current loop. The signal flow graph is shown in figure 2.2:

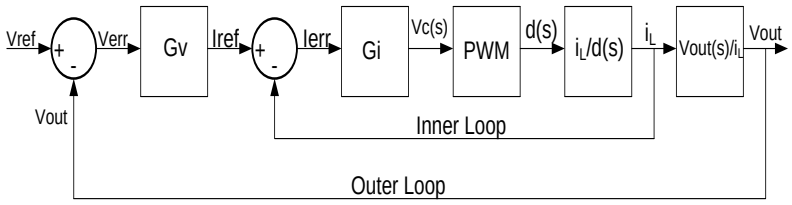

Figure 2.2: Signal Flow of Buck Converter with Current Feedback

In order to understand the system stability and transient behavior, the relationship between the input and output is required. For the inner loop, the relationship is $i_L/d(s)$. And for outer loop is $Vout/i_L$. Since the Buck converter is a nonlinear system, linearization is applied to generate the approximate linear model, which is accurate below half of the switching frequency. After applying the disturbing small signals, the formula 1.3 turns out to be:

$$V_O + \hat{v}_o = (D + \hat{d}) * (V_{IN} + \hat{v}_{in})$$

$$\hat{v}_o \approx \hat{d} V_{IN} + \hat{v}_{in} D \qquad (2.1)$$

The small signal equivalent circuit is shown below:

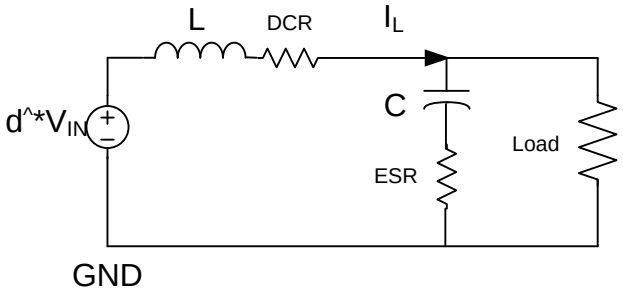

Figure 2.3: Small Signal Model of Buck Converter Power Stage

$$i_L = dV_{IN} * \frac{1}{sL + DCR + \frac{R_L(1 + sC * ESR)}{1 + sC(R_L + ESR)}}$$

$$\frac{\partial i_L}{\partial d} = \frac{V_{IN}}{R_L} * \frac{1 + sC(R_L + ESR)}{\frac{sL + DCR}{R_L} + s^2 LC(1 + \frac{ESR}{R_L}) + sDCR * C(1 + \frac{ESR}{R_L}) + (1 + sC * ESR)} \qquad (2.2)$$

$$= \frac{V_{IN}}{R_L} * \frac{1 + sC(R_L + ESR)}{s^2 LC(1 + \frac{ESR}{R_L}) + s[\frac{L}{R_L} + DCR * C(1 + \frac{ESR}{R_L}) + C * ESR] + (1 + \frac{DCR}{R_L})}$$

For the outer loop, it is easy to understand that is a current divider, inductor current will divide into two. One is load current, which is almost DC current. The other is the capacitor current, which is ripple current with DC=0.

$$\frac{\partial v_o}{\partial i_L} = \frac{R_L(1+sC*ESR)}{1+sC(R_L+ESR)} \qquad (2.3)$$

2.2. Comparison between current mode and voltage mode

- Controller Design Complexity:

  For voltage control mode, the voltage controller complexity is almost the same as current controller in current mode design. However, the advantage of the voltage controller is: the zero is load independent while current compensator has load dependence issue, especially worse when the load resistance is comparable to the capacitance equivalent resistance.

  Considering in practical design current is always required to sense for protection, extra effort is needed in voltage mode. In current mode control scheme, output current is approximately proportional to the control voltage, so it can be considered as a voltage controlled current source. In voltage loop, the transfer function is simply a one-pole system.

- Transient Response

  Current mode presents a faster transient response theoretically than voltage mode. From signal path viewpoint, voltage mode control needs to convert current signal to voltage and then compare, while current mode control can directly respond to a current change.

- Application

  This is the most attractive point of the current mode control. With current control, it is easy to achieve load sharing in parallel converters, and cycle by cycle over current protection.

2.3. Comparison between peak current mode and average current mode

- Noise Immunity

  Average current mode has a better noise immunity over peak current mode. Because in average mode, extra current compensator is added in the current loop to boost the DC gain. So high frequency noise injected in the current sensing process will not appear after the compensator. On the contrary, peak current mode samples current and translates into voltage which compares a reference directly. When the voltage is higher/lower, the switch will be on/off. So this kind of structure is very noise sensitive.

- Limitation of Peak current mode

  Peak current model is under the assumption that the current ripple is small enough and peak current can be approximate to output average current. In that way, the inductance can be ignored in transfer function calculation. However, if it is not the case, this model will have some deviation from the practical results. Besides, peak current mode is inherently instable when duty cycle exceeds 0.5. An extra compensation ramp is required. In this perspective, the complexity of these two structures is almost the same.

2.4. Design Practice

2.4.1. Power Stage

Input voltage, output voltage, output current, switching frequency, output peak-peak voltage ripple, inductor current peak-peak ripple; these kinds of parameters are always given before the design.

Suppose Vin=5V, Vo=3.3V, fs=500kHz, Iload=0.2A. With these parameter, duty cycle when stable can be calculated. D=Vo/Vin=0.66. Rload=Vo/Iload=33/2. Then assume the maximum inductor current ripple is 20% of maximum load current and maximum load current is 1A.

$$L\frac{\partial I}{\partial t}=V_{in}-V_o$$

$$L=\frac{V_O(1-D)T_S}{\Delta I_L} \qquad (2.4)$$

According to formula 2.4, L can be calculated. In this design, it is 10uH. Larger inductance gives smaller current ripples and lower conduction loss while smaller inductance has smaller size and better dynamics.

$$\Delta V_O=\frac{\Delta Q}{C}=\frac{\Delta I_L}{8Cf_s} \qquad (2.5)$$

Suppose the voltage ripple is 1%, C=15uF

ESR<0.033/0.2=0.165

2.4.2. Compensator

The voltage compensator design is based on the K-factor approach and current compensator design is based on paper conclusion.

1) Current Compensator

   In the current compensator design, 2-pole 1-zero structure can be adopted. One pole is at the origin to boost the DC and low frequency gain, a high frequency pole is placed between 1/10~1/5 DC-DC switching frequency f_s to attenuate switching noise. Since the averaging model is only valid below ½ switching frequency, the current loop unity gain frequency

cannot set too close to the switching frequency. The zero is set below the power LC filter resonant frequency1/ $\sqrt{(L \times C_L)}$ to boost phase margin and adjust the current loop crossover frequency. To make it even stable, an extra high frequency zero and pole can be added after crossover frequency. In that way, high frequency roll off will be faster.

2) Voltage Compensator

As the inner loop, current loop operates at a much faster speed than that of voltage loop. So when the voltage changes in the outer loop, the current transfer function can be considered as 1. This is the basic idea how two-loop system works. The crossover frequency of the voltage loop is set around 1/10 of the current loop crossover frequency. There is not much tricks in design voltage compensator. Standard PI compensator is suitable for design.

2.4.3. PWM Generator

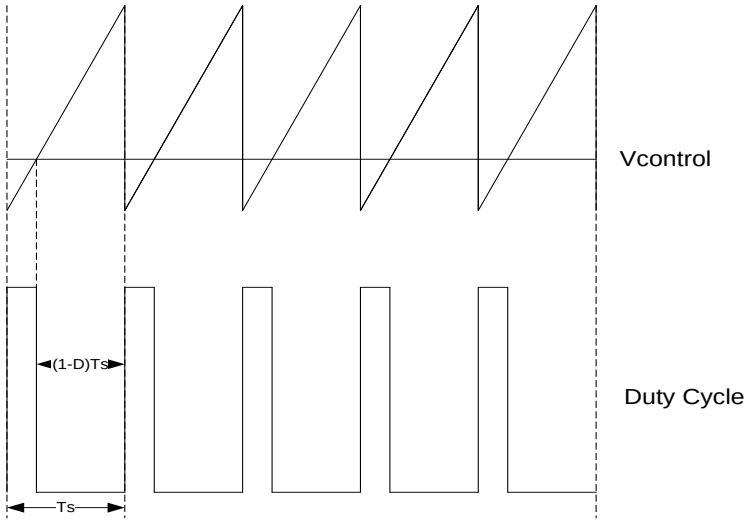

Figure 2.4: PWM & Duty Cycle

PWM generator is a comparator between the control signal and a triangular signal. When the control signal is larger than the triangular signal, the duty cycle signal is high, which means the power supply is charging the capacitance.

CHAPTER 3

SIMULINK MODEL AND SIMULATION RESULTS

3.1. Average Current Analog Model

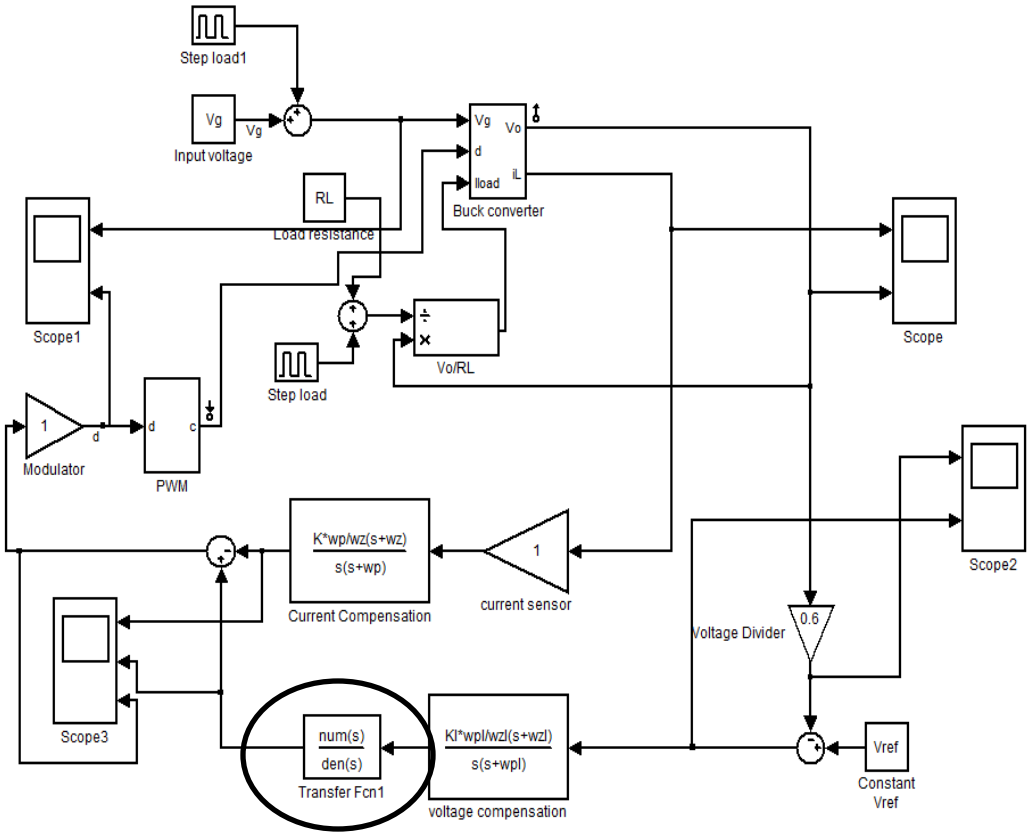

Figure 3.1: Analog Simulink Model of Buck Converter

Figure 3.1 shows the whole model of average current mode controlled Buck converter, including line regulation and load regulation. Extra attention is required of the circled function. Theoretically, the current loop is nested into the voltage loop. However, in this Simulink model, the current loop and voltage loop are implemented parallel. This transfer function is to balance the outer voltage loop.

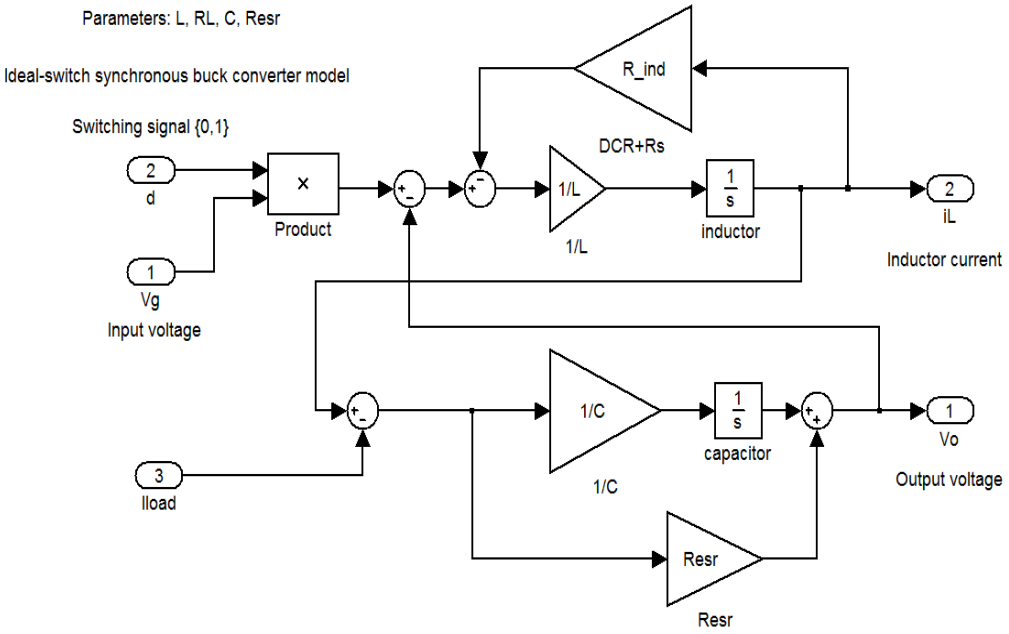

Figure 3.2: Power Stage Model

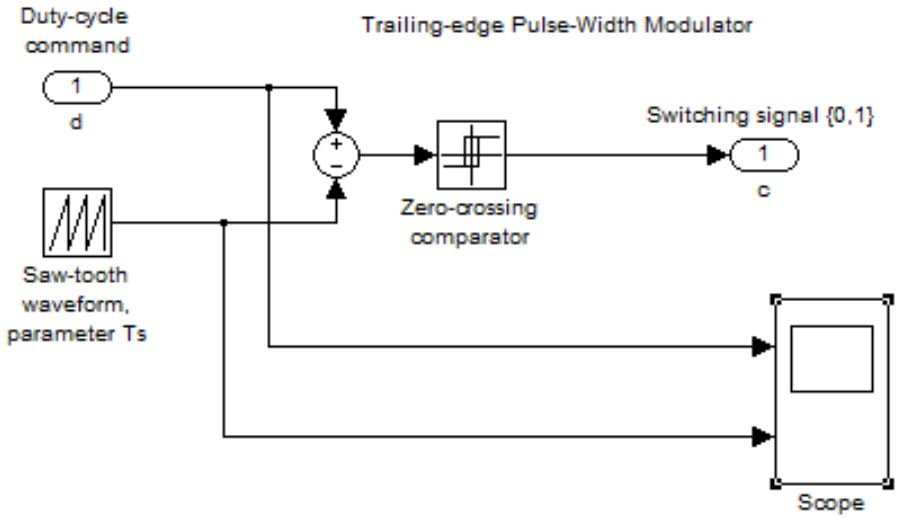

Figure 3.3: PWM Model

Main parameters used in the simulation are listed below:

Table 2: Parameters in Converter Design

Power Stage			
Input Voltage =5V	Switching frequency =500kHz	Inductance=20uH	DCR=44.8mohm
Switch resistance =250mohm	Capacitance=330uF	ESR=40mohm	Load current=0.2A
Sampling Network			
Reference Voltage=2V		Current Sensing Opamp Gain=50	
Current Compensator			
Pole=1.0368e6	Zero=9.5099e4	Gain factor=2.26e7	
Voltage Compensator			
Pole=8.4374e4	Zero=1.1686e4	Gain factor=6.092e5	

With the parameter designed in chapter 2, the bode plot of the uncompensated and compensated current loop system can be drawn.

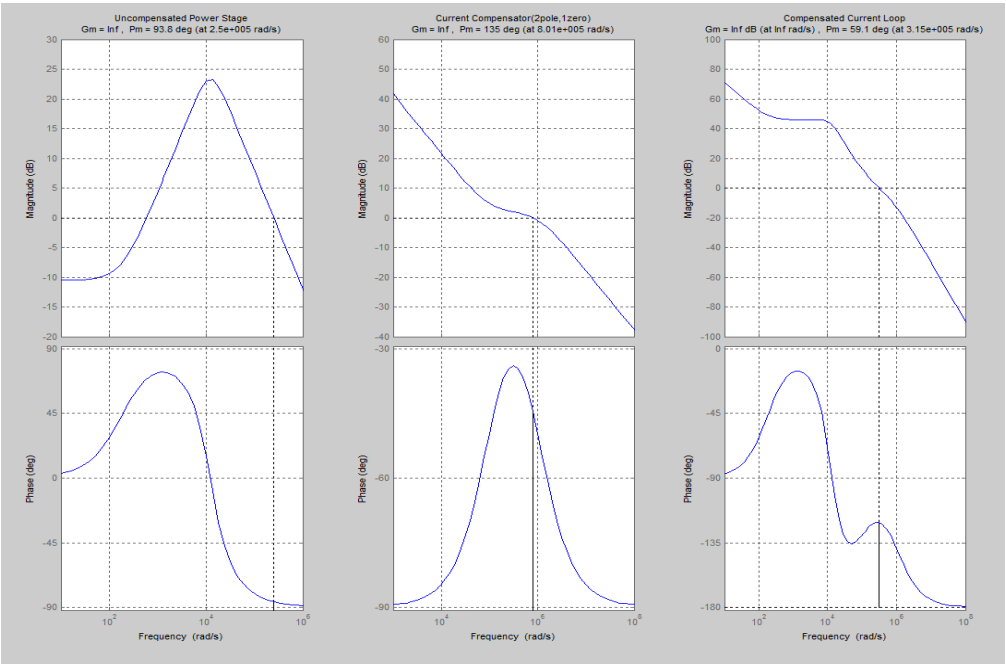

Figure 3.4: Current Loop Compensation

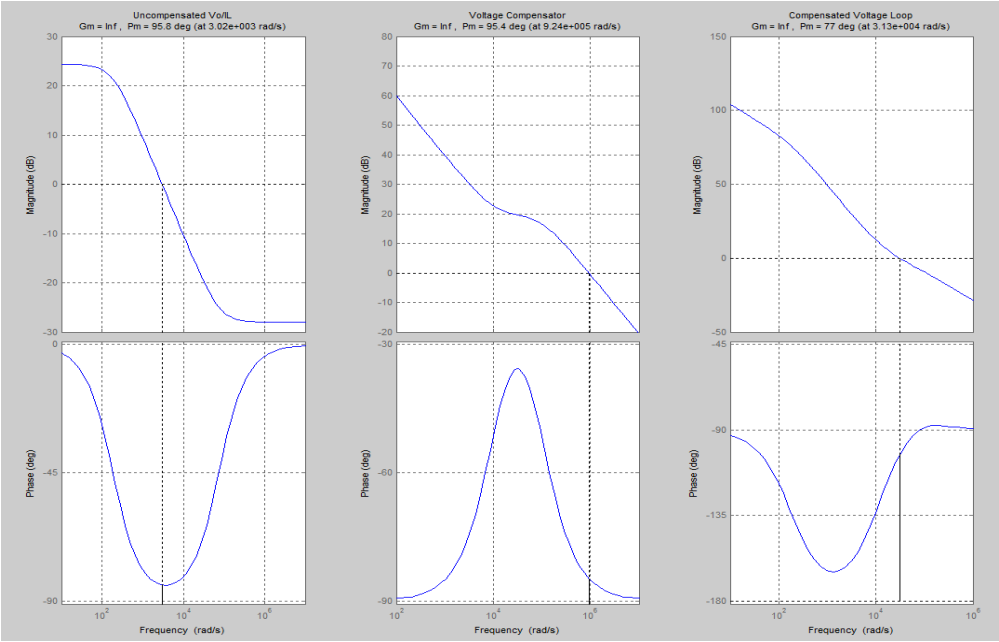

Figure 3.5: Voltage Loop Compensation

In figure 3.4 and 3.5, the left one is uncompensated system, the middle one is compensator and the right one is the compensated system.

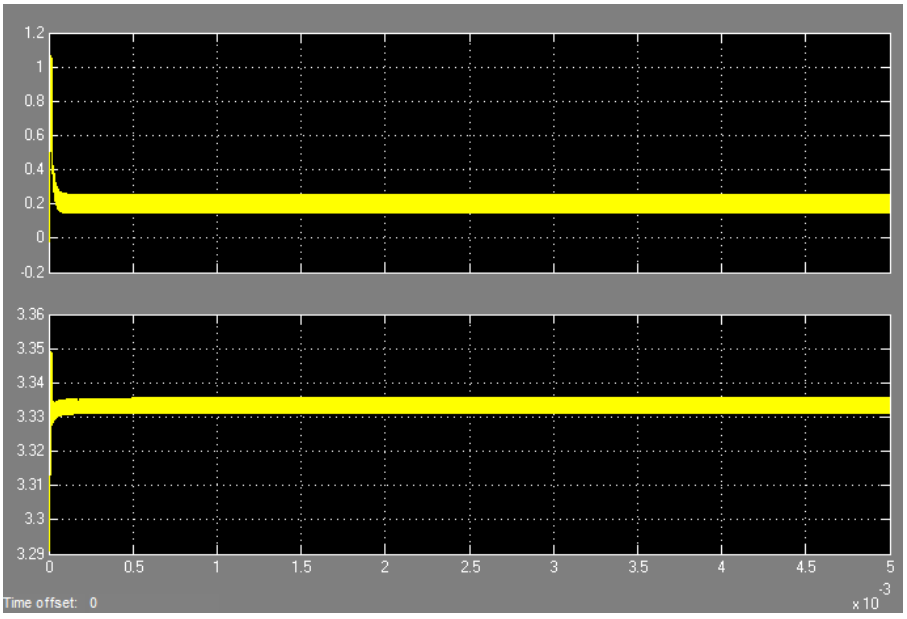

Figure 3.6: Steady State Response (3.3V, 0.2A)

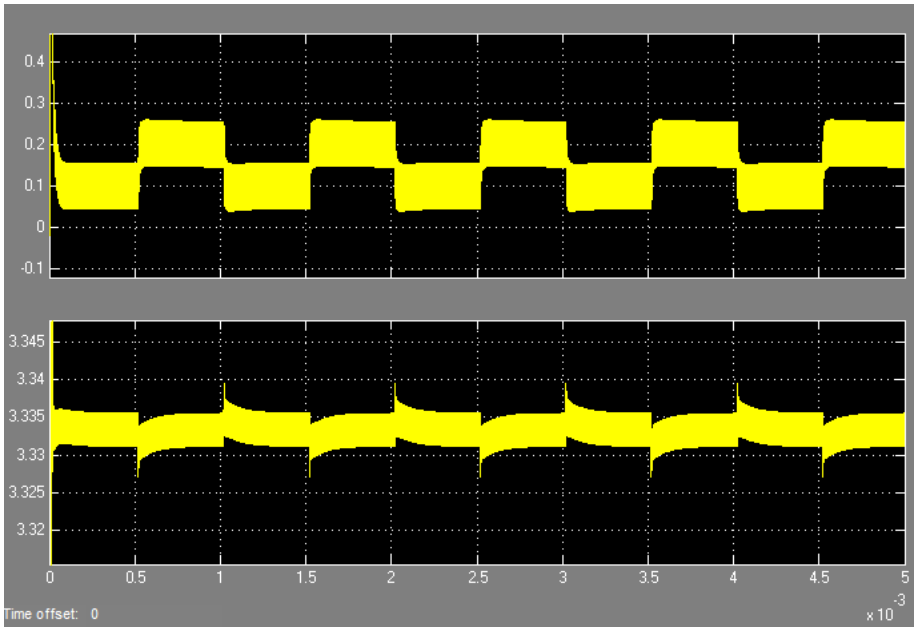

Figure 3.7: Load Regulation

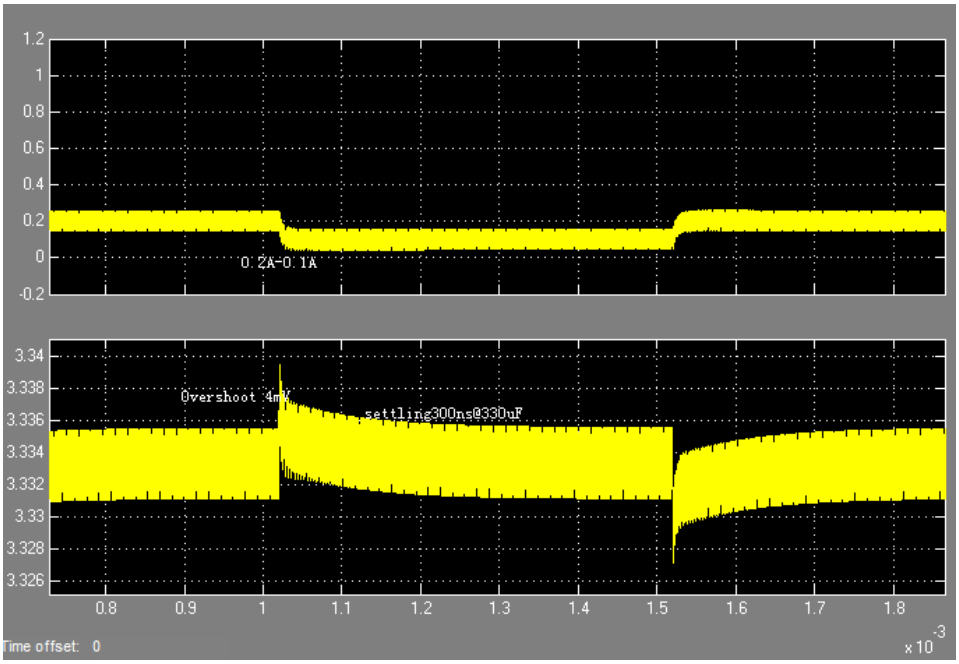

Figure 3.8: Zoom-in Load Regulation

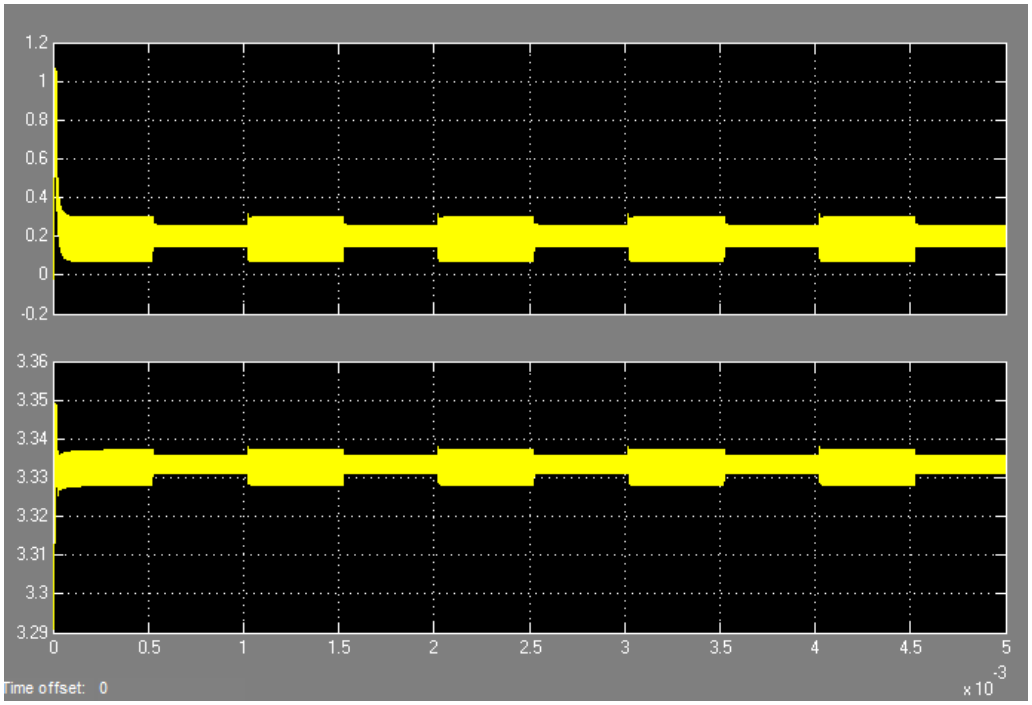

Figure 3.9: Line Regulation (5V with 40% disturbance)

3.2. Average Current Digital Model

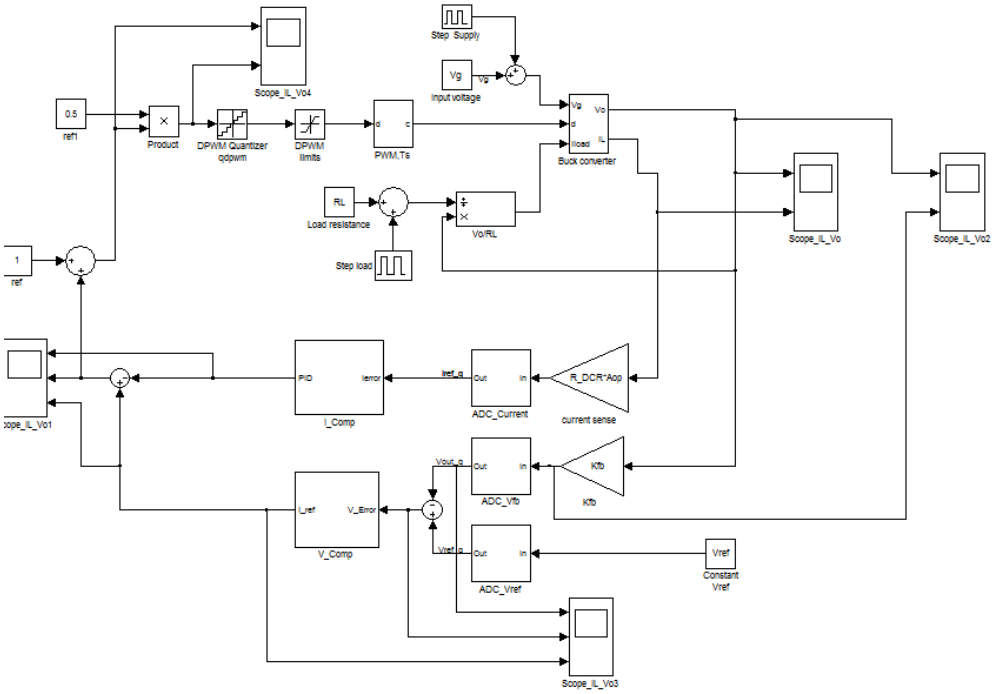

Figure 3.10: Digital Implementation of Buck Converter

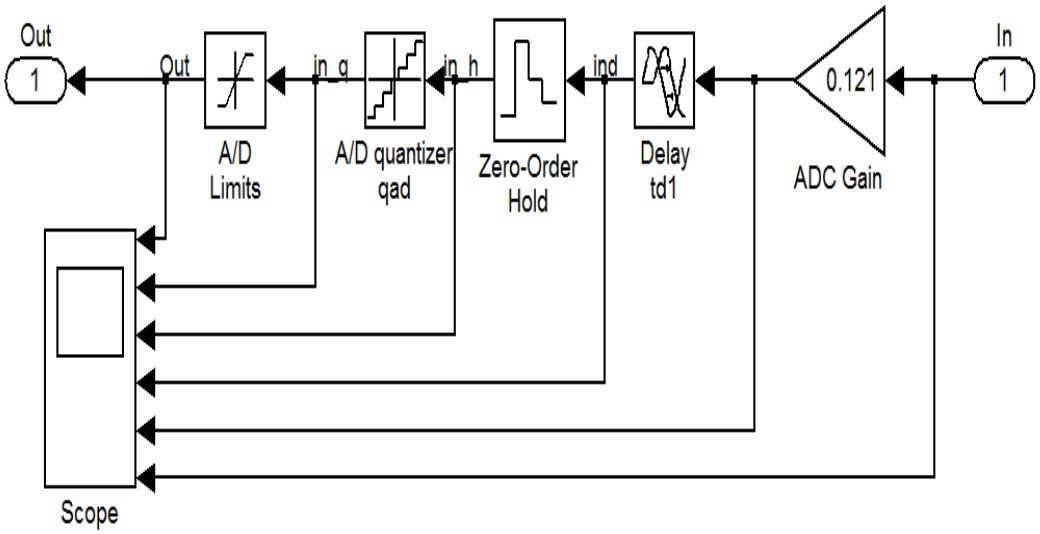

Figure 3.11: Sampling and Digitization

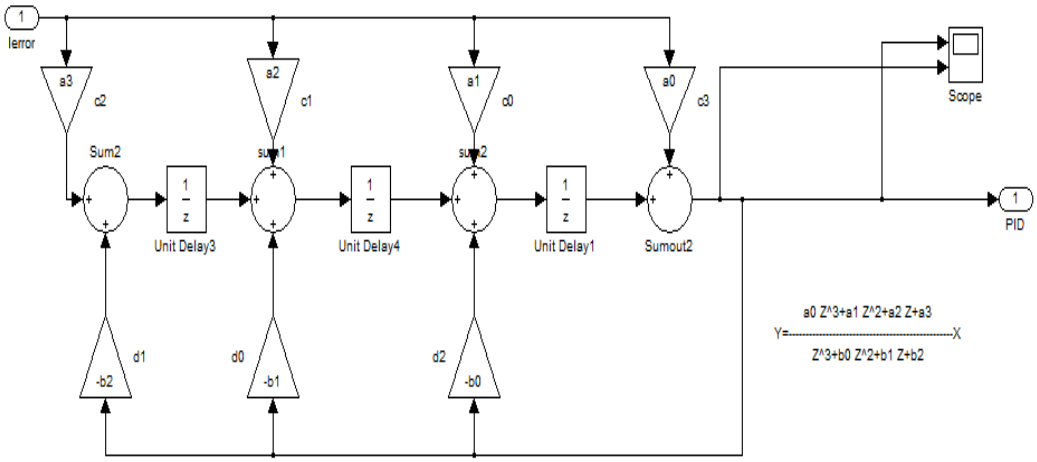

Figure 3.12 Digital Realization of Current Compensation

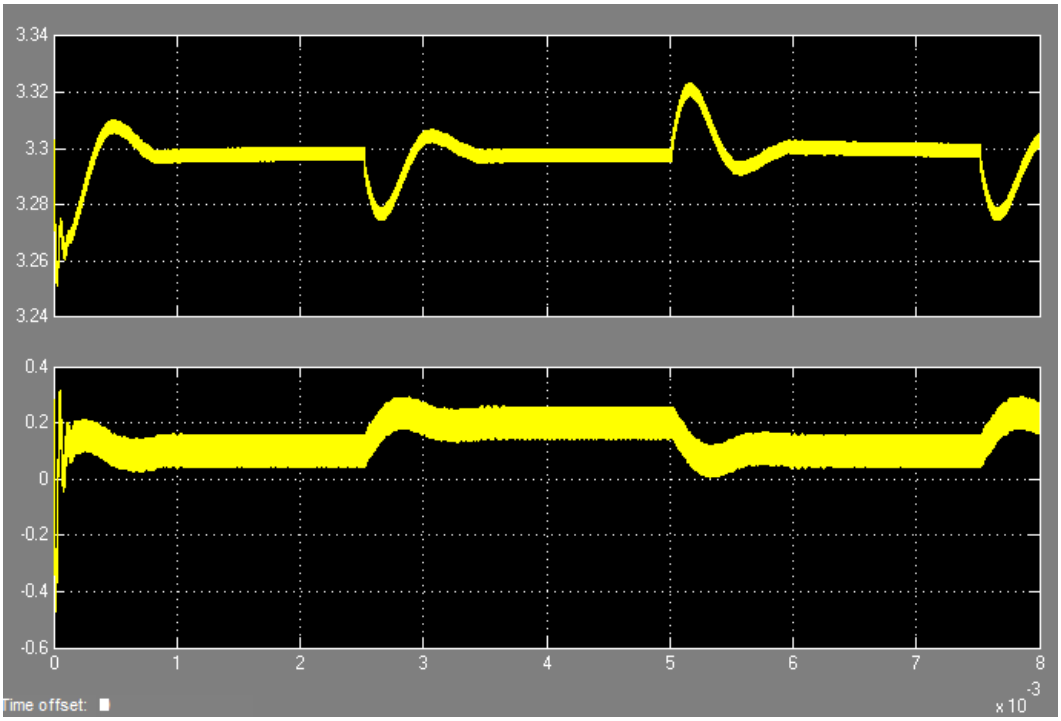

Figure 3.13 Output of Load Regulation with Digital Implementation

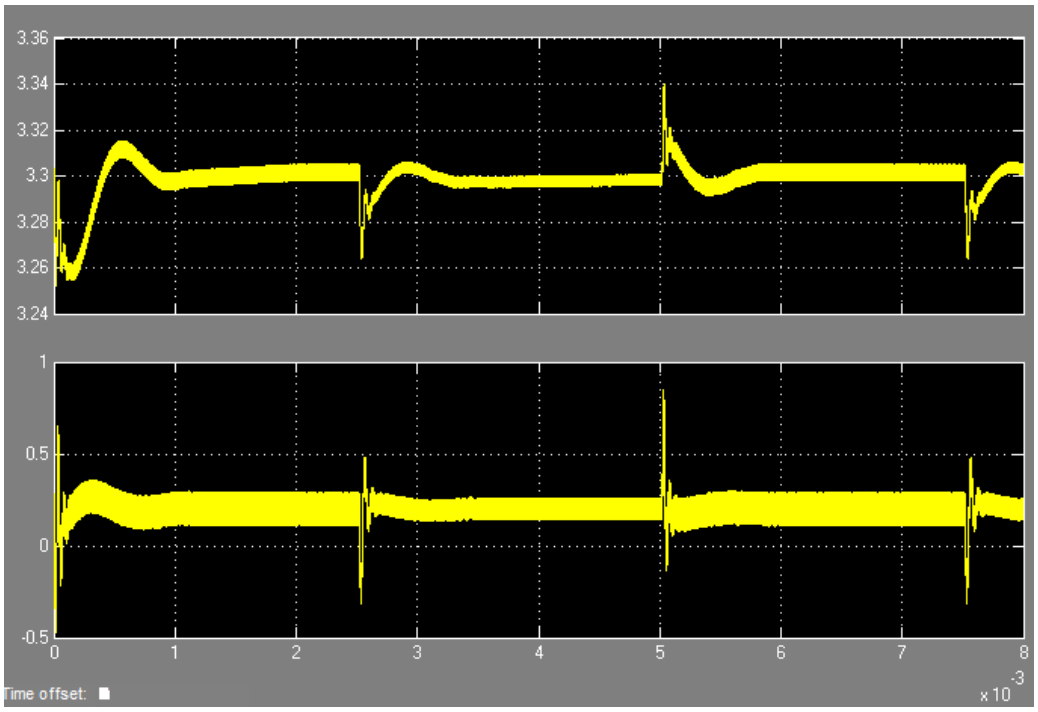

Figure 3.14: Output of Load Regulation with Digital Implementation (5V with 40% disturbance)

3.3. Efficiency Analysis

When doing load regulation including feedback loop, it is always confusing that duty cycle changes when the output maintains the same value. That is because in first order analysis, the parasitic resistance is not included in the calculation for convenience. In order to gain some thoroughly understanding of the operation of close loop Buck converter, the detailed analysis is necessary:

When the PMOS switch on, the power supply is connected to the load. The circuit structure is shown in figure 3.1. And when NMOS is on, the circuit turns out to be figure 3.2

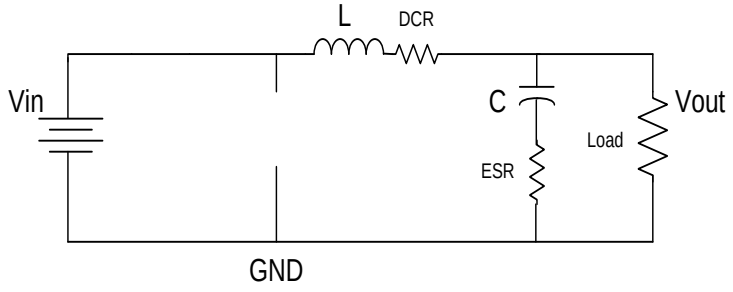

Figure 3.15: 1st Phase Buck

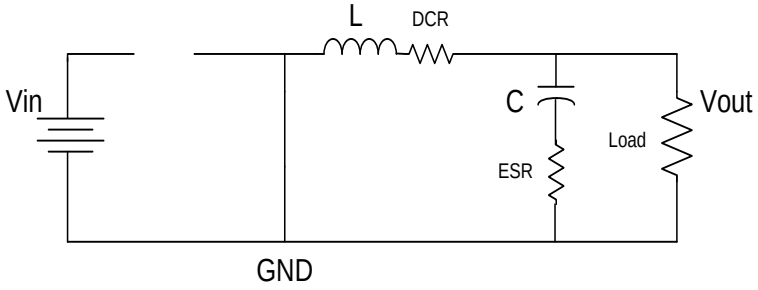

Figure 3.16: 2nd Phase Buck

During the 1st phase,

$$V_{in} = L\frac{di_L}{dt} + (R_L + R_{ds\_p}) * i_L + V_{out} \qquad (3.1)$$

Using averaging method, equation (3.1) can be simplified as

$$DT_S[V_{in} - (R_L + R_{ds\_p}) * i_L - V_{out}] = L * \Delta i_L \qquad (3.2)$$

During the 2nd phase,

$$V_{out} - L\frac{di_L}{dt} + (R_L + R_{ds\_n}) * i_L = 0 \qquad (3.3)$$

With the same way of simplification, the equation (3.3) can be transformed into:

$$(1-D)T_S[(R_L + R_{ds\_n}) * i_L + V_{out}] = L * \Delta i_L \qquad (3.4)$$

Combine the equation (3.2) and (3.4),

$$D = \frac{(R_L + R_{ds\_n}) * i_L + V_{out}}{V_{in} - (R_{ds\_p} - R_{ds\_n}) * i_L} \qquad (3.5)$$

With equation (3.5), the control system can be explained clearly. In the equation above,

R_L ---------- Inductor DCR

R_{ds_p} ---------- PMOS drain to source resistance

R_{ds_n} ----------NMOS drain to source resistance

When all these parasitic parameters are ignored, the equation (3.5) will be simplified as equation (1.3).

- Load Regulation

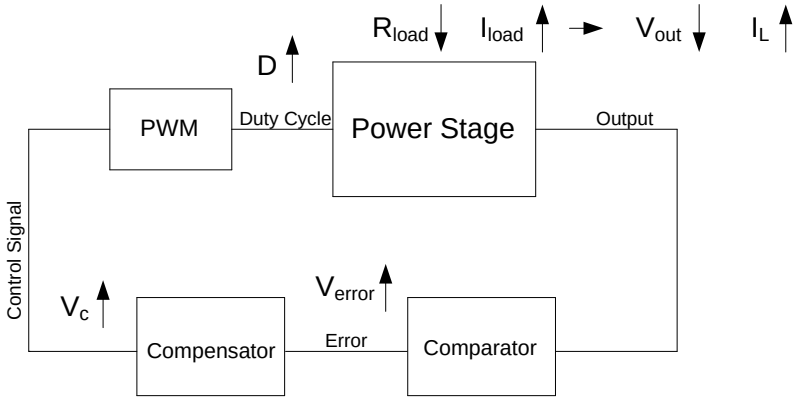

Figure 3.17: Transient Response @Load Regulation

Suppose the load resistance reduces due to the application, and load current increases since the voltage suddenly does not change. Because the load current is larger than inductor current, the capacitance has to charge the output and output voltage starts to drop. This is the point that the feedback network starts to work. At this time, the comparator compares reference with sampled output voltage and generates a positive error signal, which increases the duty cycle. As the duty cycle increases, the output voltage will increase until stable. From equation (3.5), it is also true that when inductor current goes up, the duty cycle will go up to maintain the output voltage. In the load regulation case, the output voltage change is prior to the inductor current change. Voltage feedback is faster than the current feedback with the same loop crossover frequency.

- Line regulation

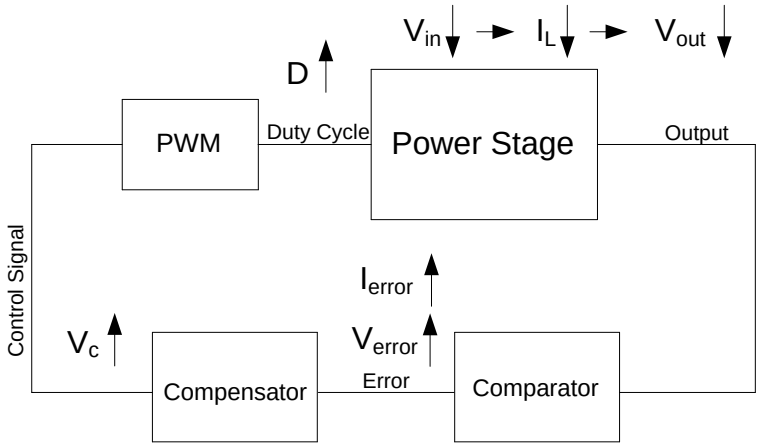

Figure 3.18: Transient Response @Line Regulation

Suppose the input voltage suffers a disturbance and becomes smaller. Then the inductor will be affected first and start to reduce. The output voltage changes after the inductor current changes, which mean the current responds faster than voltage in the line regulation. If the control loop is current feedback, the current error will increase due to the reduced sampling current which will generate an increasing duty cycle signal. If the control loop is voltage feedback, the voltage error will increase and also generate an increasing duty cycle signal. And in line regulation case, the current feedback is faster than that in voltage loop with same crossover frequency.

Based on the discussion above, the efficiency estimation can be separated in the terms below:

$$P_{L\_conduction} = R_L * [I^2_{L\_DC} + \frac{(\Delta i_L)^2}{12}] \qquad (3.6)$$

$$P_{PFET\_conduction} = R_{ds\_p} * D * [I^2_{L\_DC} + \frac{(\Delta i_L)^2}{12}] \qquad (3.7)$$

$$P_{NFET\_conduction} = R_{ds\_n} * (1-D) * [I^2_{L\_DC} + \frac{(\Delta i_L)^2}{12}] \quad (3.8)$$

$$P_{ESR} = \frac{(\Delta i_L)^2}{12} * ESR \quad (3.9)$$

From equation (3.6) to (3.9), it can be seen that the power loss is directly related to the peak-to peak current ripple, which is determined by the value of inductor. Besides, as the switching frequency increase, the power loss will also increase. Since the PFET and NFET will charge and discharge parasitic capacitance, such as Cgs, Cgd, during the switching period.

CHAPTER 4

DESIGN FLOW OF CONTROLLER IN FPGA

4.1. Introduction

In order to digitize the analog sensing output, frequency discriminator is used as the Analog-to Digital converter. Also the controller needs to be digitized. In this design, the decimator and feedback compensator is integrated in FPGA.

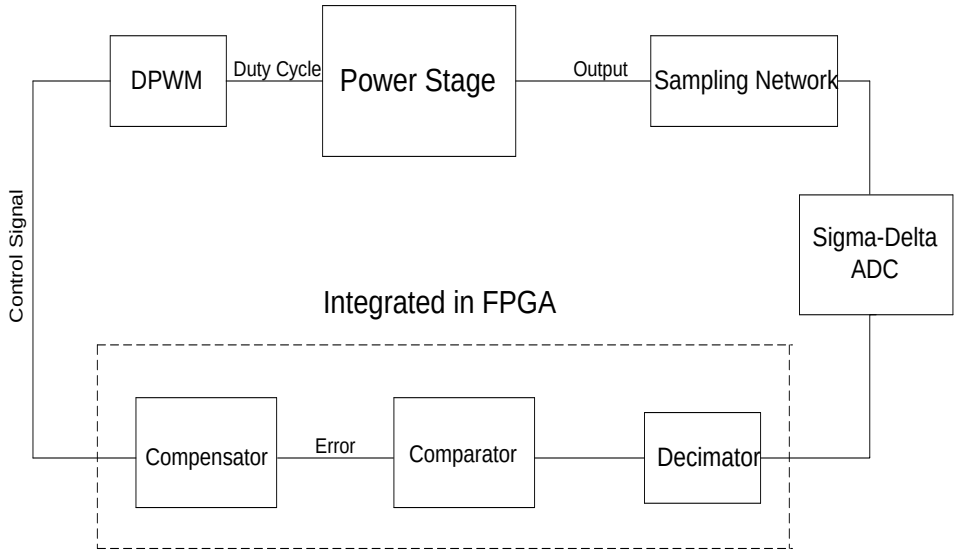

Figure 4.1: FPGA Integration

The output signal of sampling network is a voltage. The voltage difference passes a voltage controlled oscillator and generates a frequency changeable one-bit stream, which is the input of the decimator. Through the decimator, the signal turns out to be a multi-bit stream. And finally after algorithm calculations, the control signal sent to the DPWM is a 9-bit signal. The bit-width of the control signal is determined by the DPWM design which is not covered in the thesis. The FPGA implementation with fixed digit flow is shown in figure 4.2:

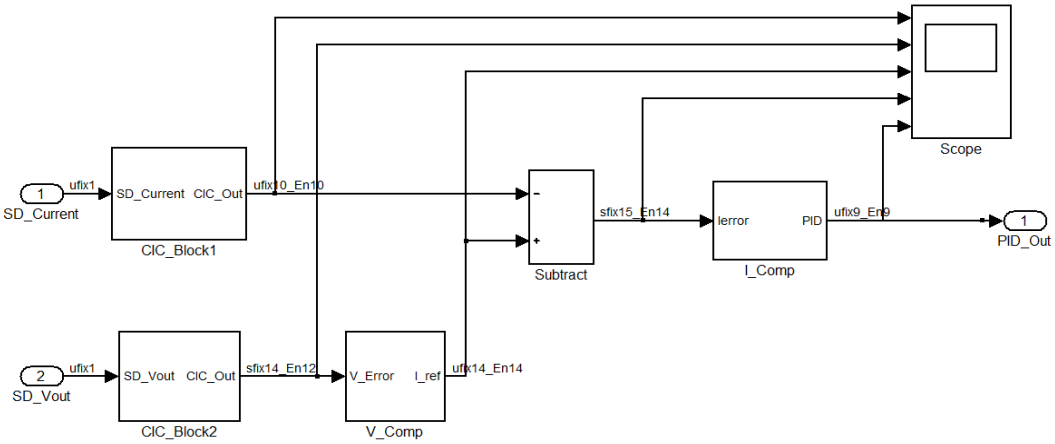

Figure 4.2: FPGA Flow

4.2. Decimation

In the control loop, actually three variables need to be digitized: reference voltage, sampled output voltage and sampled inductor current. Since the reference voltage is a constant value, this variable can be tested individually and pre-store in the FPGA in Look-up table format.

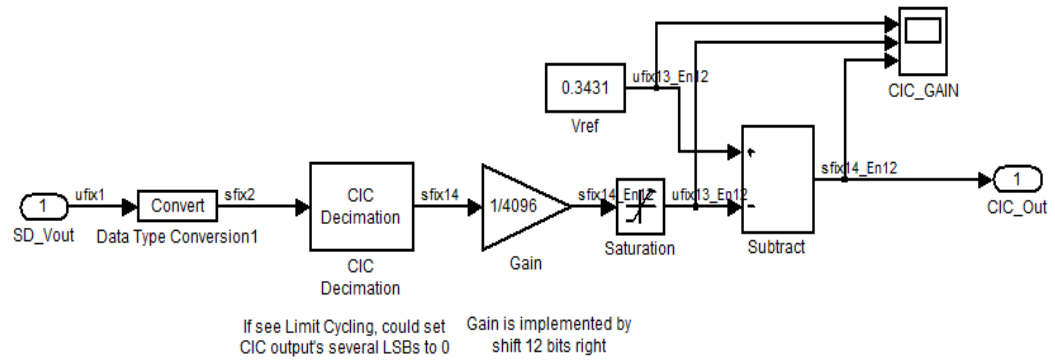

Figure 4.3: Decimation

Figure 4.3 shows the details of the CIC_Block in figure 4.2. When 1-bit input is sent into decimator, it generates an output with 14-bit signed signal.

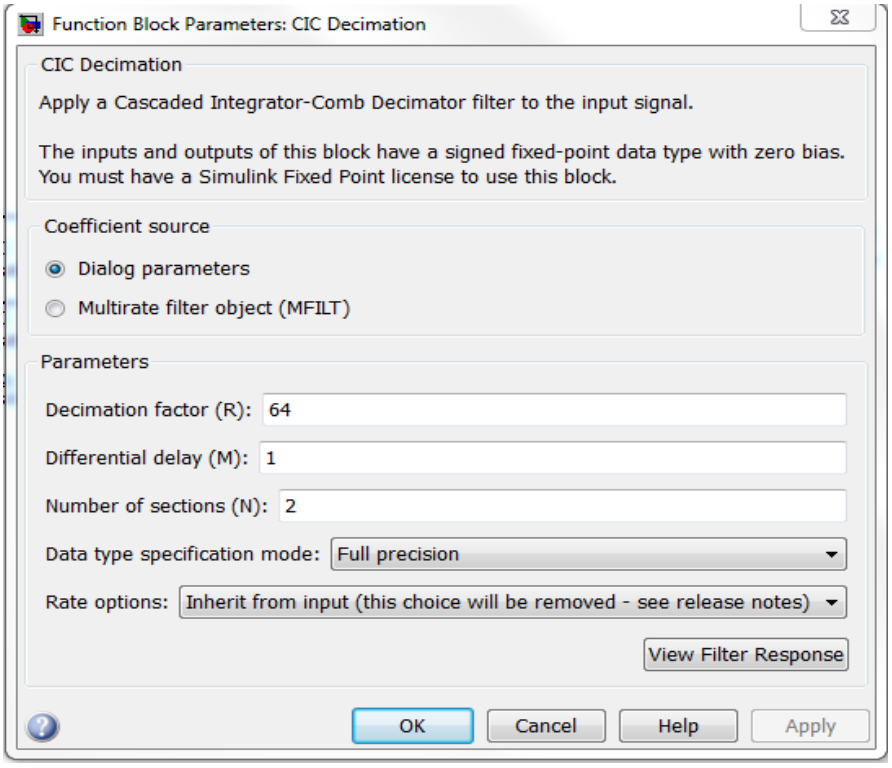

Figure 4.4: Decimator Parameter Setting

The CIC decimation parameter setting is shown above. CIC (Cascaded Integrator Comb) filter is used to extract narrow band signal from a wideband sources. Decimator factor is the frequency ratio between the input and output. In this case, the frequency of silicon oscillator is 32MHz and the switching frequency is 500kHz. So the decimation factor is 32/0.5=64. M is a design parameter which can be any positive integer, but it is usually limited to 1 or 2 []. N refers to the number of cascaded comb stages. With these three parameters, the output bit length could be decided:

$$B_{out} = N \log_2 RM + B_{in} \qquad (4.1)$$

In this design, output bit length is 14. The reason why to right shift the output by 12 bits is:

In VCO design, the full range of ADC output is 0—4095. In order to normalize the output, 1/4096 is multiplied.

4.3. Compensator

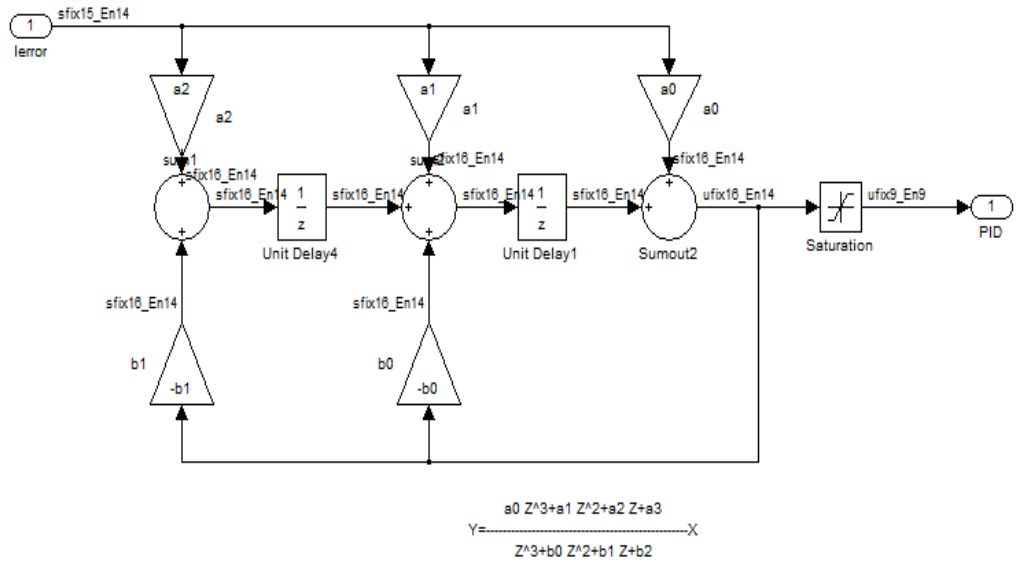

Figure 4.5: Fixed Digit Compensator

The parameter of compensator in FPGA is basically the same as that in chapter 2. The tricky part is the bit length selection. Since this is a digital implementation, the bit length determines the LSB which is the accuracy of the whole system. And the bit length cannot be too large. Larger bit length means more implementation units and slower it can operate. It is the tradeoff between the speed and accuracy. Every Multiplication will increase or reduce the bit length depends on the value. And every addition may increase the bit length. So the bit length decision requires careful consideration.

4.4. Simulation Results

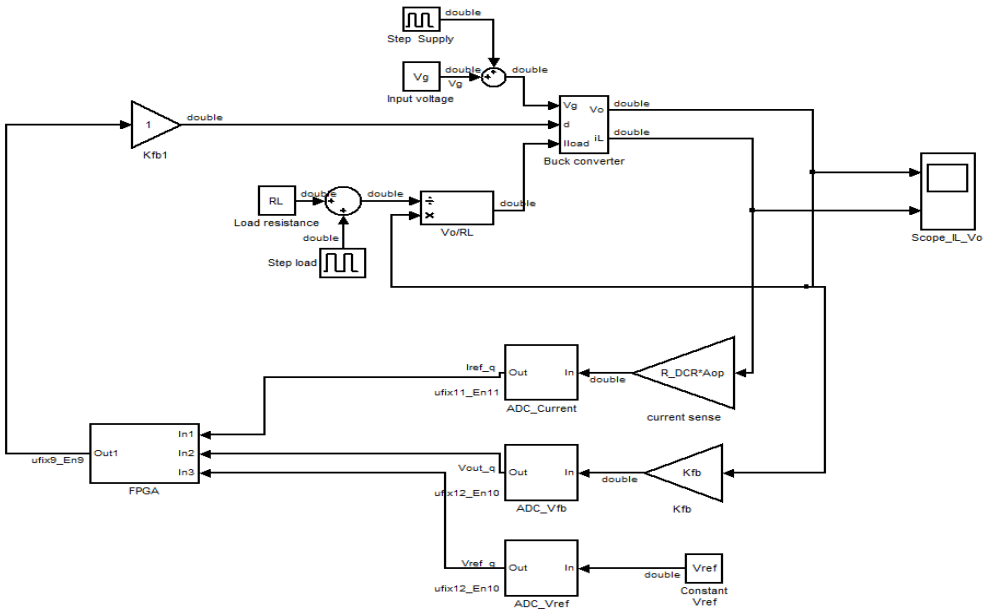

Figure 4.6: Simulink Model with Fixed Digit Signal

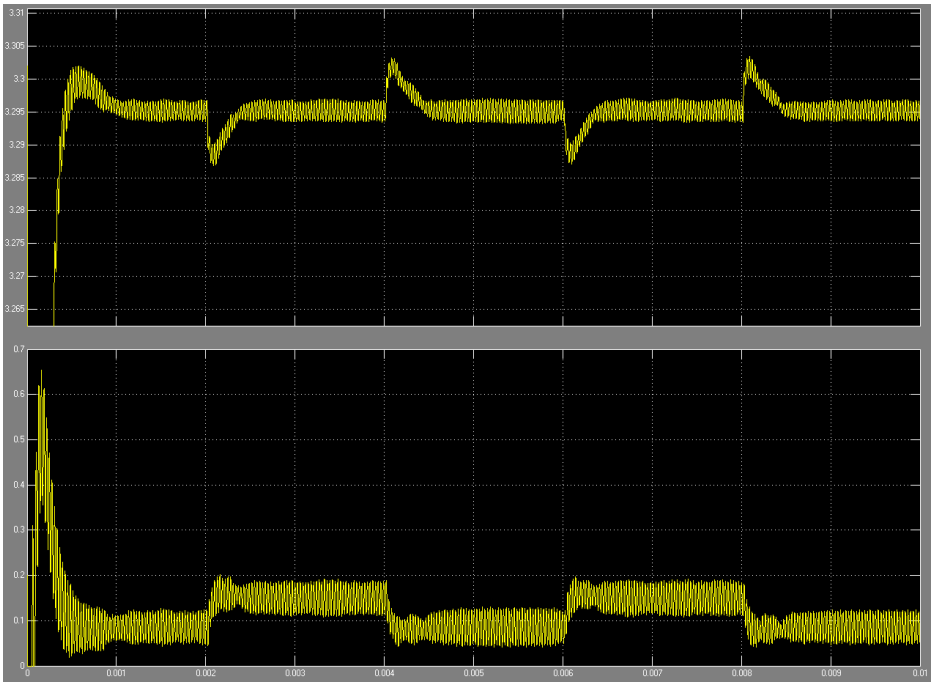

Figure 4.7: Load Regulation with fixed Digit Compensator

In figure 4.6, all the signals are implemented in fixed digit. The compensator is implemented in FPGA manner. In this way, the simulation can have the maximum

similarity with the board test. Figure 4.7 is the simulation result of the load regulation. The voltage ripple is around 3mV. When load jumps, the output voltage overshoot is around 10mV. Settling time is approximately 500us.

CHAPTER 5

BOARD TEST RESULTS

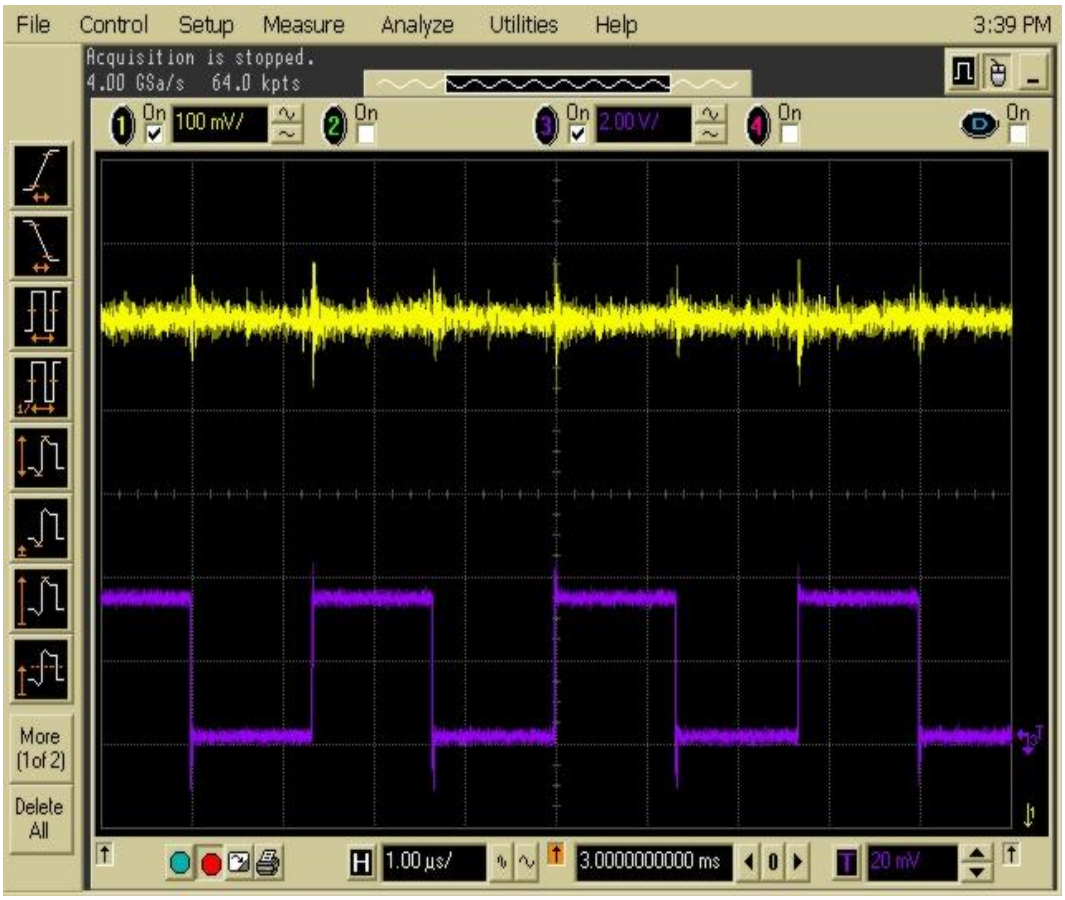

Figure 5.1: Output Voltage & PWM

The clock frequency is 24MHz, input voltage is 5V and output voltage is 2.4V with load resistance 33ohm.

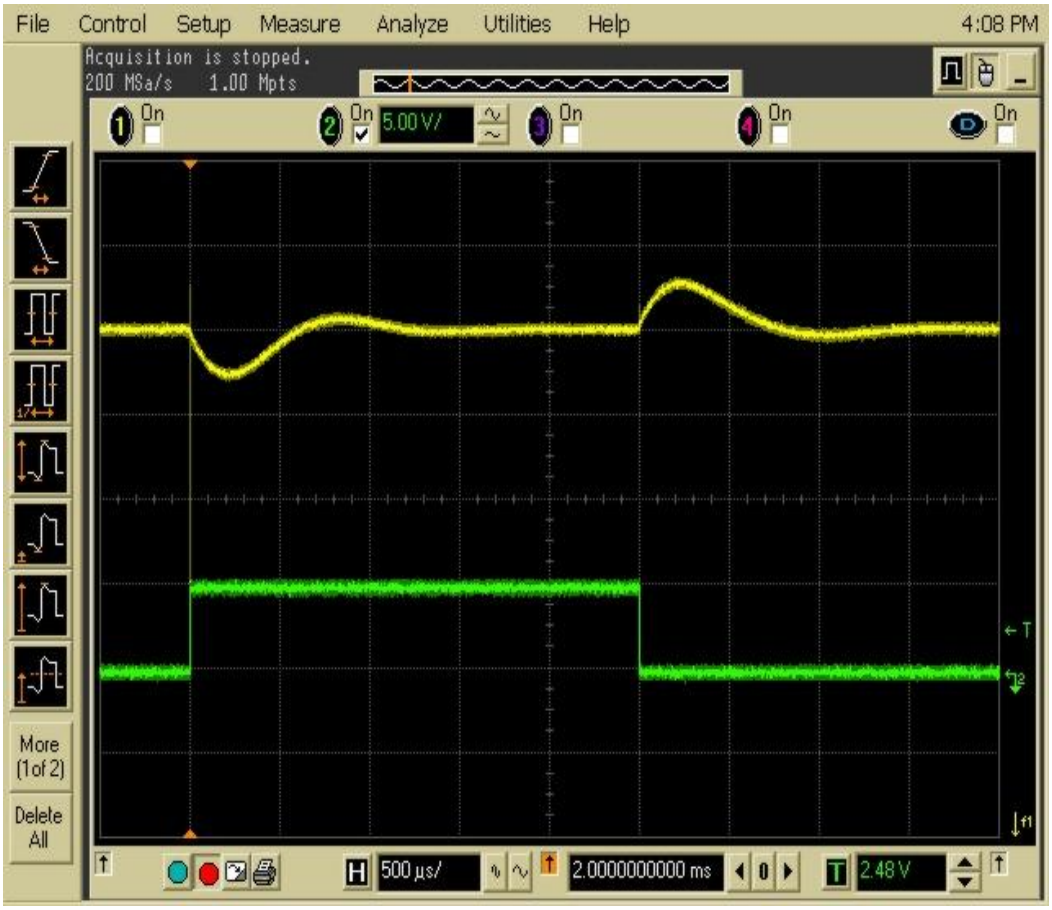

Figure 5.2: Load Regulation @CLK=23MHz

The clock frequency is 23MHz. Input voltage is 5V and output is 2.2V. Load switching is from 16.5ohm to 11ohm. The load switching frequency is 200Hz.

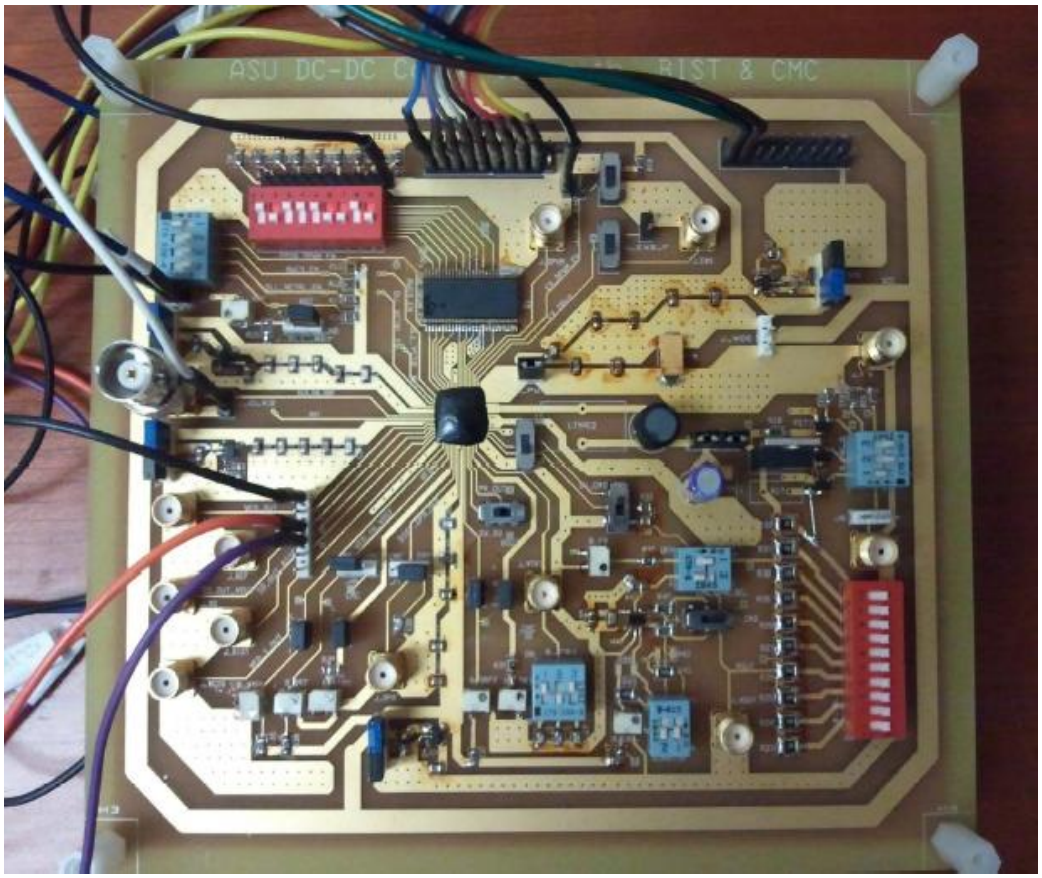

Figure 5.3: Test Board Graph

CHAPTER 6

SUMMARY

This design practice is based on the design of voltage loop controlled Buck converter on chip. The current loop control is added on top of the voltage loop converter. The comparison of the two kinds of control loop design will improve the understanding of the DC-DC converter design.

Although in board test, the speed advantage of current mode over voltage mode is not obvious, current mode has broad vision in parallel applications. Since the parallel battery supply has its benefit in extending the battery lifetime, the unique advantage of current control will has its best platform:

- Current addition is cost-free
- Over current protection is an inherit characteristic of current control

In these kinds of applications, current mode control will show the superiority over voltage mode.

REFERENCES

[1] S.Castillo, N.K. Samala, K. Manwaring, B.Izadi and D.Radhakrishnan, "Experimental Analysis of Batteries under Continuous and Intermittent Operations", ESA, pp. 18-24, Jun. 2004

[2] Tao Liu, Hyunsoo Yeom, Vermeire, B, Adell, P, Bakkaloglu, B, "A digitally controlled DC- DC buck converter with lossless load-current sensing and BIST functionality", ISSCC Dig. Tech. Papers, pp. 388 – 390, Feb. 2011.

[3] H. Ahmad, B. Bakkaloglu, "A 300mA 14mV-Ripple Digitally Controlled Buck Converter Using Frequency Domain ΔΣ ADC and Hybrid PWM Generator," ISSCC Dig. Tech. Papers, pp. 202-203, Feb. 2010.

[4] M. Ilic, D. Maksimovic, "Digital Average Current-Mode Controller for DC–DC Converters in Physical Vapor Deposition Applications," IEEE Tran. Power Electronics, vol. 23, pp. 1428 - 1436, May. 2008.

[5] J. Chen, A. Prodic, R. W. Erickson, et al., "Predictive Digital Current Programmed Control," IEEE Trans. Power Electron., vol. 18, pp. 411–419, Jan. 2003.

[6] Wei Tang, Fred C. Lee, and Raymond B. Ridley, "Small Signal Modeling of Average Current Mode Control", IEEE Trans. On Power Electronics, Vol. 8, No.2, April 1993, pp.112-119.

APPENDIX A

AVERAGE CURRENT MODE CONTROL SIMULINK SCRIPT

(ANALOG)

```
%-----------------------------------------------------------------
%
% The model parameters is set for "buck_analog_control.mdl"
simulink
% module.
%
% Run the script first to set all the parameters.
%
%-----------------------------------------------------------------
clc;close all; clear;
% Power stage parameters
Vg=5;   % input voltage
fs=500e3;  % switching frequency
Ts=1/fs;
L=20e-6;  % filter inductance
R_ind=44.8e-3+250e-3; % equivalent resistance of inductor
C=330e-6;  % filter capacitance
Resr=40e-3;  % equvalent serial resistance
RL=33/2;  % load resistance
Iload=0.2;
Vref=2;  % reference voltage

% current sensing component
Rs=1; % sensing resistor
Acl=1; % sensing gain

% Modulator
Vm=1.6; % peak voltage of sawtooth
Km=1/Vm; % modulator gain

% uncompensated current loop
num_power_current=[C*(RL+Resr),1];
den_power_current=[L*C*(1+Resr/RL),L/RL+R_ind*Resr*C/RL+C*R_ind+C
*Resr,1+R_ind/RL];
Ho_power_current=tf(num_power_current, den_power_current);
H_dc=Ho_power_current*Vg/RL;

% current compensation
[Gm1,Pm1,Wg1,Wp1] = margin(H_dc)
Phi1=60;
Phase_comp=Phi1+(180-Pm1)-90;
Theta=(Phase_comp/2+45)/180*3.14;
kc=tan(Theta)
wz=fs*2*3.14/(10*kc);
wp=(fs*2*3.14/10)*kc;
num_cc=[1,wz];
den_cc=[1,wp,0];
Ho_cc=tf(num_cc, den_cc);
[gain1,ph1]=bode(Ho_cc, 314e3)
k=120499;
K=k*wp/wz;
H_cc=K*Ho_cc
```

```
%compensated current loop
H1=H_cc*H_dc
[Gm2,Pm2,Wg2,Wp2] = margin(H1)
figure(1)
margin(H1)
% grid

% uncompensated voltage loop
num_power_voltage=[C*(RL*Resr),RL];
den_power_voltage=[C*(RL+Resr),1];
Ho_power_voltage=tf(num_power_voltage, den_power_voltage);
H_power_voltage=Ho_power_voltage;

% voltage compensation
[Gm3,Pm3,Wg3,Wp3] = margin(H_power_voltage)
Phi2=60;
Phase_compl=Phi2+(180-Pm3)-90;
Thetal=(Phase_compl/2+45)/180*3.14;
kcl=tan(Thetal)
wzl=fs*2*3.14/(100*kcl);
wpl=(fs*2*3.14/100)*kcl;
num_vc=[1,wzl];
den_vc=[1,wpl,0];
Ho_vc=tf(num_vc, den_vc);
[gain2,ph2]=bode(Ho_vc, 314e2)

Kl=kl*wpl/wzl;
H_vc=Kl*Ho_vc

%compensated voltage loop
H2=H_vc*H_power_voltage
[Gm4,Pm4,Wg4,Wp4] = margin(H2)
figure(2)
margin(H2)
```

(DIGITAL)

```
%-------------------------------------------------------------------
-
%
% This is a parameter file that sets the parameters for the
% "buck_CMC_Digital.mdl" simulink model
%
% Please run this script first to set the necessary variables
%
%-------------------------------------------------------------------
-
clc;close all; clear;

%-------------------------------------------------------------------
-
%  Set up system and components parameters
```

```
%-------------------------------------------------------------------
-
% Power stage parameters
Vg=5;   % input voltage
fs=500e3;  % switching frequency
Ts=1/fs;
L=20e-6;  % filter inductance
R_DCR=44.8e-3; % Inductor DCR
R_ind=R_DCR+250e-3; % equivalent resistance of inductor+ Power
MOSFET Ron
C=330e-6;  % filter capacitance
Resr=40e-3;  % equivalent serial resistance of cap
Ts = 1/fs; % switching period
Iload=0.2;
Vref=2;  %  Reference Voltage
Kfb=Vref/3.3; %  Vout feedback ratio
RL = (Vref/Kfb)/Iload; % Load resistor

% current sensing component
% Isense= R_DCR*Aop;
Aop=50; % sensing OPA gain

% A/D, DPWM and coefficient quantization parameters
qad = 1/2^9*0.121; % LSB of the A/D converter 8bits A/D
ndpwm = 9; % number of DPWM bits
qdpwm = 1/2^ndpwm; % DPWM resolution

% td: total sampling, computing, and modulator delay, td = td1 +
DTs
td1 = 4.5e-6;  % td1: delay from the sample instant to the rising
edge of PWM

%-------------------------------------------------------------------
-
% Construct compensators
%-------------------------------------------------------------------

% uncompensated current loop
num_power_current=[C*(RL+Resr),1];
den_power_current=[L*C*(1+Resr/RL),L/RL+R_ind*Resr*C/RL+C*R_ind+C
*Resr,1+R_ind/RL];
Ho_power_current=tf(num_power_current, den_power_current);
H_dc=Ho_power_current*Vg/RL /(R_DCR*Aop) ;
subplot(2,3,1);
bode(H_dc);
title('1. Uncompensated current loop H_dc');
grid
% current compensation
[Gm1,Pm1,Wg1,Wp1] = margin(H_dc)
Phi1=65;
Phase_comp=Phi1+(180-Pm1)-90;
Theta=(Phase_comp/2+45)/180*3.14;
kc=tan(Theta)
wz=fs*2*3.14/(10*kc);
```

```
wp=(fs*2*3.14/10)*kc;
num_cc=[1,wz];
den_cc=[1,wp,0];
Ho_cc=tf(num_cc, den_cc);
num_cc_c=[1,1e6];
den_cc_c=[1,1e7];
H_cc_c=tf(num_cc_c, den_cc_c);
[gain1,ph1]=bode(Ho_cc, 314e3)
k=1/gain1*2;
K=k*wp/wz;
H_cc=K*Ho_cc*H_cc_c
subplot(2,3,2);
bode(H_cc);
title('2. Current composentor H_cc'); grid

%compensated current loop
H1=H_cc*H_dc
[Gm2,Pm2,Wg2,Wp2] = margin(H1)
subplot(2,3,3);
margin(H1)
title('3. compensated current loop H1');

% uncompensated voltage loop
num_power_voltage=[C*(RL*Resr),RL];
den_power_voltage=[C*(RL+Resr),1];
Ho_power_voltage=tf(num_power_voltage, den_power_voltage);
H_power_voltage=Ho_power_voltage;
subplot(2,3,4);
bode(H_power_voltage)
title('4. uncompensated voltage loop');
grid

% voltage compensation
[Gm3,Pm3,Wg3,Wp3] = margin(H_power_voltage)
Phi2=55;
Phase_compl=Phi2+(180-Pm3)-90;
Thetal=(Phase_compl/2+45)/180*3.14;
kcl=tan(Thetal)
wzl=fs*2*3.14/(100*kcl);
wpl=(fs*2*3.14/100)*kcl;
num_vc=[1,wzl];
den_vc=[1,wpl,0];
Ho_vc=tf(num_vc, den_vc);
[gain2,ph2]=bode(Ho_vc, 314e2)
kl=1/gain2;

Kl=kl*wpl/wzl;
H_vc=Kl*Ho_vc
subplot(2,3,5);
bode(H_vc)
title('5. Voltage composentor H_vc');
grid

H_comp=H_cc+1;
```

```
H=H_comp*H_vc;
%compensated voltage loop
H2=H_vc*H_power_voltage
[Gm4,Pm4,Wg4,Wp4] = margin(H2)
subplot(2,3,6);
margin(H2)
title('6compensated voltage loop H2');

%-----------------------------------------------------------------
-
% Generate digital compensators coefficients
%-----------------------------------------------------------------
-
% Convert to Z domain
H_cc_d=c2d(H_cc,Ts,'tustin')
n_cc=get(H_cc_d,'num')
n=n_cc{1};
a0=n(1);
a1=n(2);
a2=n(3);
a3=n(4);

d_cc=get(H_cc_d,'den')
d=d_cc{1};
b0=d(2);
b1=d(3);
b2=d(4);

H_d=c2d(H,Ts,'tustin')
n_H=get(H_d,'num')
n=n_H{1};
c0=n(1);
c1=n(2);
c2=n(3);
c3=n(4);
c4=n(5);
c5=n(6);

d_H=get(H_d,'den')
d=d_H{1};
d0=d(2);
d1=d(3);
d2=d(4);
d3=d(5);
d4=d(6);

al=20; % a coeff total length
af=12; % a coeff fraction length
bl=20; % b coeff total length
bf=12; % b coeff fraction length
```

APPENDIX B

VHDL SCRIPT

```
--  -------------------------------------------------
----------
--
-- Module: FPGA
-- Source Path: PID_V1/FPGA
-- Hierarchy Level: 0
--
--  -------------------------------------------------
----------
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;
USE  IEEE.std_logic_unsigned.ALL;  ---Added  library  for
math operation


ENTITY FPGA IS
  PORT( CLK                         :   IN    std_logic;
        RST                         :   IN    std_logic;
        CLK_EN                      :   IN    std_logic;
        SD_Current_IN               :   IN    std_logic;
-- ufix1
        SD_Vout_IN                  :   IN    std_logic;
-- ufix1
        CE_OUT                      :   OUT   std_logic;
        PID_Out  :OUT  std_logic_vector(8  DOWNTO  0)   --
ufix9_En9
         );
END FPGA;

ARCHITECTURE rtl OF FPGA IS

  -- Component Declarations
  COMPONENT FPGA_tc
    PORT( CLK                                 :    IN
std_logic;
          RESET                               :    IN
std_logic;
          CLK_EN                              :    IN
std_logic;
          enb_1_1_1                           :    OUT
std_logic;
          enb_1_64_0                          :    OUT
std_logic;
          enb_1_64_1                          :    OUT
std_logic
```

```
          );
  END COMPONENT;

  COMPONENT FPGA_Core
    PORT( CLK                                     :    IN
std_logic;
          RESET                                   :    IN
std_logic;
          enb_1_1_1                               :    IN
std_logic;
          enb_1_64_0                              :    IN
std_logic;
          SD_Current              :   IN    std_logic;
-- ufix1
          SD_Vout                 :   IN    std_logic;
-- ufix1
          PID_Out: OUT std_logic_vector(8 DOWNTO 0) --
ufix9_En9
          );
  END COMPONENT;

  -- Component Configuration Statements
  FOR ALL : FPGA_tc
    USE ENTITY work.FPGA_tc(rtl);

  FOR ALL : FPGA_Core
    USE ENTITY work.FPGA_Core(rtl);

  -- Signals
  SIGNAL enb_1_1_1                        : std_logic;
  SIGNAL enb_1_64_0                       : std_logic;
  SIGNAL enb_1_64_1                       : std_logic;
  SIGNAL FPGA_Core_out : std_logic_vector(8 DOWNTO 0);
-- ufix9

  SIGNAL PID                       : std_logic_vector(8
DOWNTO 0);
  SIGNAL MSB                       : std_logic_vector(4
DOWNTO 0);
  SIGNAL LSB                       : std_logic_vector(3
DOWNTO 0);

  SIGNAL SD_Vout_D1                         : std_logic;
-- ufix1
  SIGNAL SD_Vout                            : std_logic;
-- ufix1
```

```
  SIGNAL SD_Current_D1                      : std_logic;
-- ufix1
  SIGNAL SD_Current                         : std_logic;
-- ufix1

  SIGNAL RST_D1                            : std_logic;
  SIGNAL RST_D2                            : std_logic;
  SIGNAL RST_D3                            : std_logic;
  SIGNAL RESET                             : std_logic;

BEGIN
  u_FPGA_tc : FPGA_tc
    PORT MAP( CLK => CLK,
              RESET => RESET,
              CLK_EN => CLK_EN,
              enb_1_1_1 => enb_1_1_1,
              enb_1_64_0 => enb_1_64_0,
              enb_1_64_1 => enb_1_64_1
              );

  u_FPGA_Core : FPGA_Core
    PORT MAP( CLK => CLK,
              RESET => RESET,
              enb_1_1_1 => enb_1_1_1,
              enb_1_64_0 => enb_1_64_0,
              SD_Current => SD_Current_IN,  -- ufix1
              SD_Vout => SD_Vout_IN,  -- ufix1
              PID_Out => FPGA_Core_out1  -- ufix9_En9
              );

    CE_OUT <= enb_1_64_1;

  ------------------------------
  -- RESET Signal
  ------------------------------
  RESET <= RST_D3 AND RST_D2 AND RST_D1 AND RST; ---
to eliminate reset noises.
  --RESET <='0';
  PROCESS (CLK)
  BEGIN
      IF ( CLK'EVENT AND  CLK='1') THEN
          RST_D1 <= RST;
          RST_D2 <= RST_D1;
          RST_D3 <= RST_D2;
      END IF;
```

```
  END PROCESS;

  ------------------------------
  -- PID
  ------------------------------
  PID <= FPGA_Core_out1;
  --PID_Out <= PID;
  PID_Out <= MSB&LSB;
  PROCESS (PID)
  BEGIN
      IF PID(3 DOWNTO 0)="1111" THEN
          LSB <="1111";
      ELSE  LSB <=PID(3 DOWNTO 0);
      END IF;
      IF PID(8 DOWNTO 4)="11111" THEN
          MSB <="11111";
      ELSE  MSB <=PID(8 DOWNTO 4) + "00001";
      END IF;
  END PROCESS;

  ------------------------------
  -- SDFD bit stream read
  ------------------------------
  PROCESS (CLK,RESET)
  BEGIN
      IF RESET ='1' THEN
          SD_Vout <= '0';
                SD_Current <='0';
      ELSIF ( CLK'EVENT AND  CLK='1') THEN
          SD_Vout_D1 <= SD_Vout_IN;
                SD_Vout    <= SD_Vout_D1;
          SD_Current_D1 <= SD_Current_IN;
                SD_Current    <= SD_Current_D1;

      END IF;
  END PROCESS;


END
rtl;
```