A Low Power Digital Controller for DC-DC Converter Applications with Integrated PFM Mode Detector

by

Abhiram Mumma Reddy

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

Approved September 2014 by the
Graduate Supervisory Committee:

Bertan Bakkaloglu, Chair
Umit Ogras
Jae-sun Seo

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

Switching Converters (SC) are an excellent choice for hand held devices due to their high power conversion efficiency. However, they suffer from two major drawbacks. The first drawback is that their dynamic response is sensitive to variations in inductor (L) and capacitor (C) values. A cost effective solution is implemented by designing a programmable digital controller. Despite variations in L and C values, the target dynamic response can be achieved by computing and programming the filter coefficients for a particular L and C. Besides, digital controllers have higher immunity to environmental changes such as temperature and aging of components. The second drawback of SCs is their poor efficiency during low load conditions if operated in Pulse Width Modulation (PWM) mode. However, if operated in Pulse Frequency Modulation (PFM) mode, better efficiency numbers can be achieved. A mostly-digital way of detecting PFM mode is implemented. Besides, a slow serial interface to program the chip, and a high speed serial interface to characterize mixed signal blocks as well as to ship data in or out for debug purposes are designed. The chip is taped out in 0.18μm IBM's radiation hardened CMOS process technology. A test board is built with the chip, external power FETs and driver IC. At the time of this writing, PWM operation, PFM detection, transitions between PWM and PFM, and both serial interfaces are validated on the test board.

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Bertan Bakkaloglu, for giving me this opportunity to work on this project and for supporting me financially. I would also like to thank Prof. Umit Ogras and Prof. Jae-sun Seo for taking time to serve as panel members on my defense committee.

I am fortunate to have worked with good team players for this project. A special thanks to Navankur and Debasis for their help in evaluating the chip. I am grateful to my friends in office who made it an enjoyable and learning place. I would also like to thank James Laux for his IT support.

Finally, I would like to thank my parents, my brother and friends for their support.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

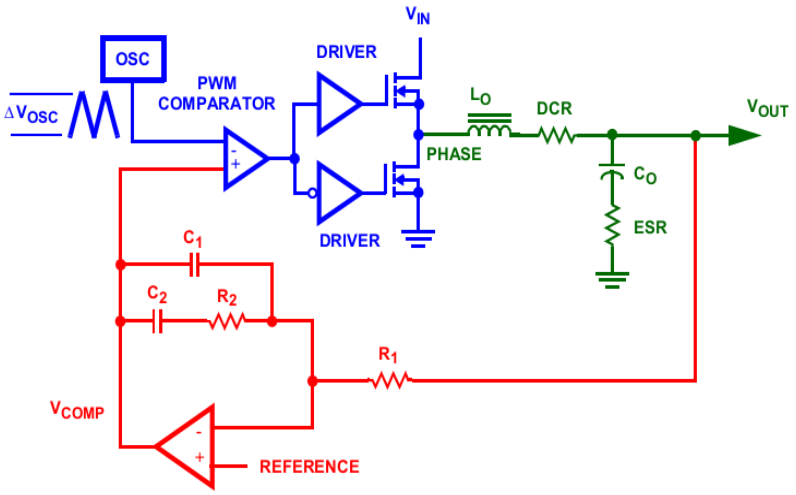
Figure Page

1 Introduction

1.1 Background

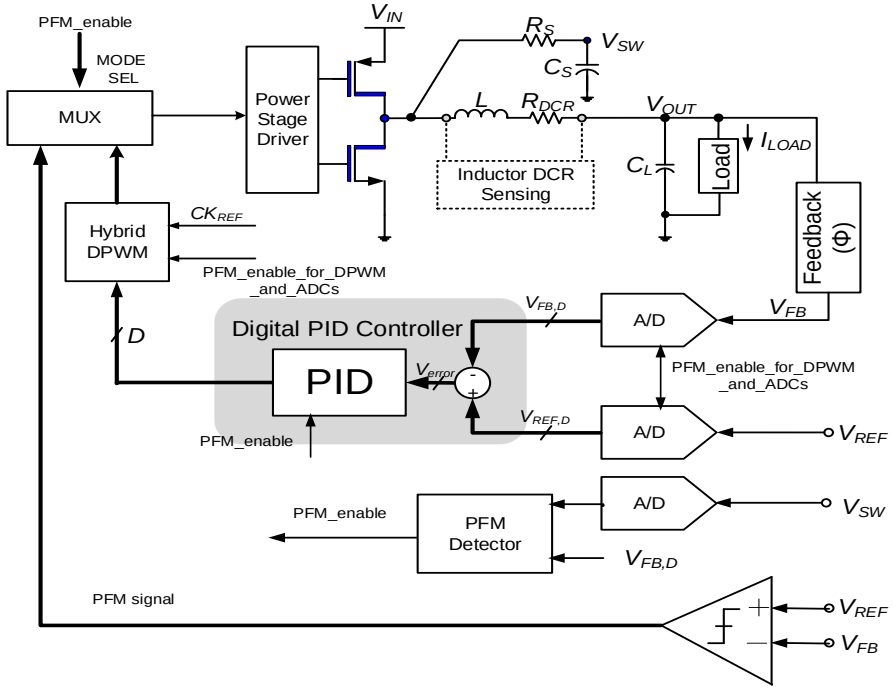
One day or the other, most of us would have been victims of our handheld device running out of battery. Situations like these emphasize the importance of battery life in a handheld device. In a broad sense, this problem is attacked with three methodologies. First, through innovations in battery technologies like increase in battery density, second, through efficient power distribution in a device, third, through application based architecture level optimizations. The topic of interest in this thesis is efficient power distribution methodology. Here, to deliver power to various blocks in a device, Switching Converters (SC) are used to deliver high power where as Low Dropout Regulators (LDO) are used to deliver low power. However, SCs have two major drawbacks. First, they need external inductor (L) and capacitor (C), whose variations in values affect the dynamic response of the SC. Second, they offer poor efficiency at low loads.

The first problem can be solved by choosing precision inductor and capacitor which is not cost effective. This thesis presents a solution where in a programmable digital controller is designed to account for variations in L and C values. Shown in the next page is a typical Buck SC. In the Figure 1, components shown in blue and red are inside the chip, and in green are external to chip. Based on the values of L, C, DCR and ESR, the software like Matlab computes the filter coefficients to achieve the target dynamic response. To ship the computed coefficients into the chip, a Serial Peripheral Interface is designed.



1. PWM Based Switching Converter

The second problem is solved to some extent by operating the SC in Pulse Frequency Modulation (PFM) mode at low loads. This thesis presents a way to detect PFM mode using mostly digital logic. Shown below is the architecture.



2. PFM Detection Architecture

As illustrated in the 2, the averaged switching node voltage (V_{SN}) is digitized, and in PFM Detector block, digitized output voltage is subtracted from V_{SN} yielding the DC drop across L (V_{DCR}) since L has finite DC Resistance (R_{DCR}). Assuming R_{DCR} to be constant, V_{DCR} - a measure of load current, is compared against a programmable value to decide on entering into PFM mode. The control logic that implements this functionality is designed.

In addition to the aforementioned blocks, a high speed serial interface is designed to characterize ADC and DPWM blocks as well as to ship out / in data for debug purposes. The Buck Controller with these blocks is taped out in 0.18µm IBM's radiation hardened process technology.

1.2 Thesis Organization

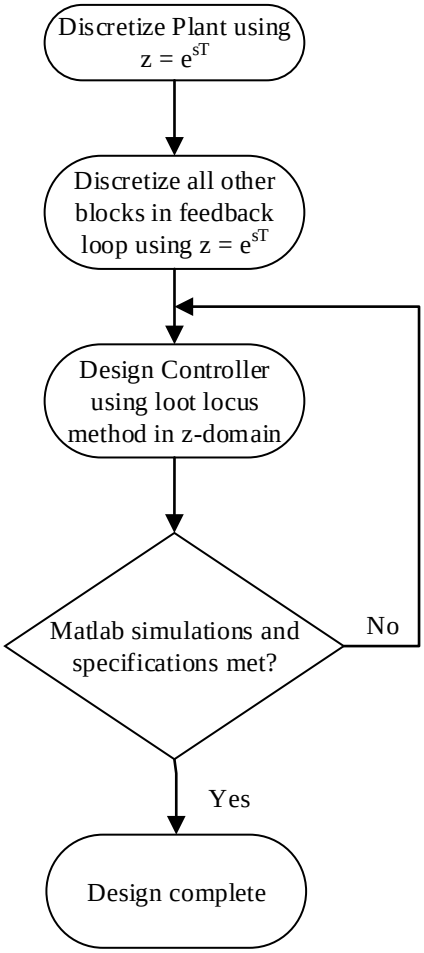
In chapter 2, existing design methods are described, and the chapter ends with reasons supporting the choice of a particular design method. The chapter 3 deals with the modeling and design of Controller and PFM Detector in Matlab for given specifications. Chapter 4 covers the digital implementation details of all digital blocks that were put on chip. The results obtained are presented in chapter 5 followed by conclusions and future work in chapter 6.

2 Controller Design Methods

The Controller design methods can be classified into two types namely z-domain method and s-domain method. All of these methods are explained below.

2.1 Controller Design using z-domain Method

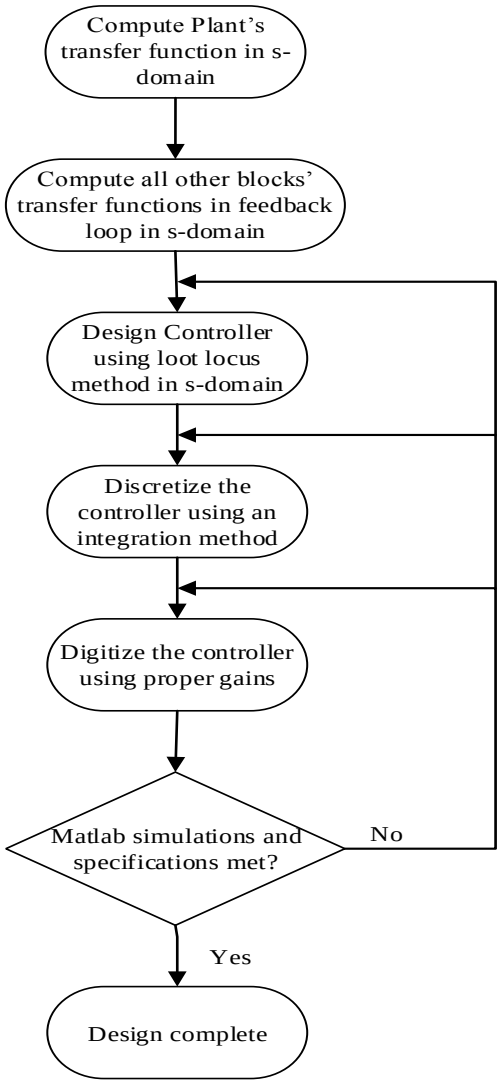
In this method, Controllers are designed using the root locus method in z-domain. The first step is to discretize the Plant (LC stage). The continuous time transfer function of the Plant is discretized using the matched pole-zero mapping method where Plant's poles and zeros are mapped to the z-plane according to the equation $z = e^{sT}$. Here, T is the sampling time with which the loop is clocked. Similarly, transfer functions of all other blocks in the loop are obtained. Once loop's transfer function is obtained in z-domain, controller is designed using root locus method where poles and zeros are placed in order to achieve the target dynamic response. Typically, high level software like Matlab is used for the design. The design procedure is summarized in the Figure 3 in the next page.



3. Flow Chart for Controller Design in Z-Domain

2.2 Controller Design using s-domain Method

In this method, Controllers are designed using the root locus method in s-domain. The Plant's transfer function is computed in s-domain. Besides, s-domain transfer functions of all other blocks in the loop are obtained. Based on the loop's characteristics observed in the Bode plot, Controller is designed by placing poles and zeros to achieve the required phase margin, gain margin and settling time. The obtained Controller in s-domain is then discretized using any of the integration methods. The design procedure is summarized in the Figure 4 in the next page.



4. Flow Chart for Controller Design in S-Domain

2.3 Choosing a Design Method

For Controller designs, following differences are observed between z-domain and s-domain methods.

z-domain method	s-domain method
Plant and all other continuous time blocks are discretized, in other words	None of the blocks other than controller is discretized

approximated	
Controller is designed based on the loop function created from above approximate models	Controller is designed using exact continuous time domain transfer functions
Gives comparable performance with s-domain method in some cases only as mentioned in [5]	Guaranteed performance match with that of s-domain controller if enough sampling rate is chosen which is usually the case
Extra work in discretizing all continuous time models in loop	Only once the controller is discretized once it's s-domain counterpart is ready

1. Differences between Z-Domain and S-Domain Methods

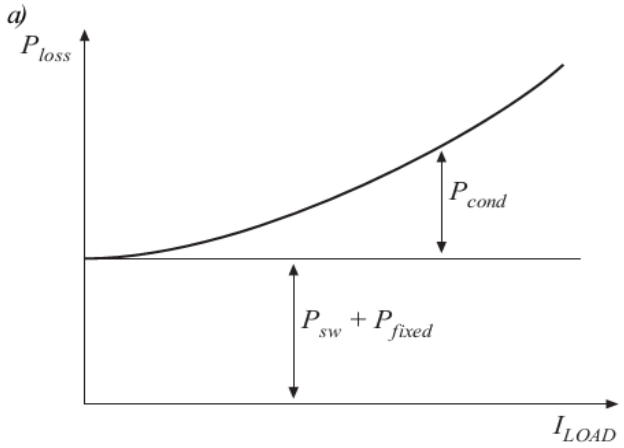
In the light of above observations, s-domain method is chosen for designing the controller and the step by step procedure is described in chapter 4.

3 PFM Operation and Detection

3.1 Need for PFM

To reduce power consumption, all handheld devices come with a Sleep/Standby mode, where the current consumption ranges from few µA to mA. Switching Converters when operated with fixed frequency suffer from poor efficiencies at these load currents, because switching losses start becoming significant. Also, typically, the efficiency of the converter in PWM mode drops below 60% at approximately 500mA load current as mentioned in [10].

To understand the need for PFM, it is worth noting the sources of various losses in the Switching Converters. The figure below shows the distribution of losses in a fixed frequency converter.

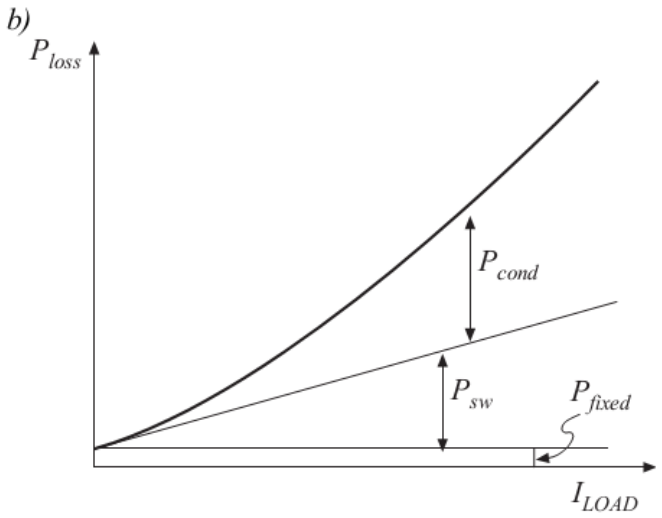


5. Variation of Losses in a Fixed Frequency Converter

As illustrated in above figure, a conventional Switching Converter has three types of power losses namely fixed losses (P_{fixed}), switching losses (P_{sw}) and conduction losses (P_{cond}). The total power loss in a switching power converter can be expressed as

$$P_{loss} = P_{fixed} + P_{sw} + P_{cond}$$

The fixed losses do not depend on load current or switching frequency. They are the power consumed by all the circuits that form a feedback loop. The switching losses are due to turning on and off of the switching MOSFET's. They heavily depend on the switching frequency. The conduction losses are due to the power loss in MOSFETs and Inductor Resistance. These scale with the load current i.e. lower the load lower the conduction losses and vice versa. Given the nature of the above losses, the only way to improve the efficiency of the Switching Converter at low loads is to make the P_{sw} scalable with load current as shown below.

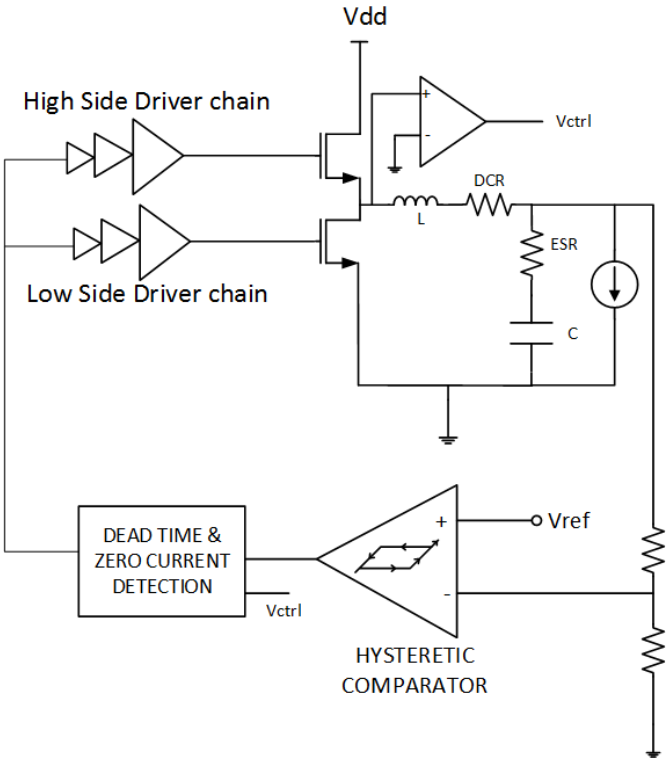


6. Variation of Losses in a Variable Frequency Converter

If frequency of the signal driving the MOSFETs is made scalable is with load current, then the switching losses can be reduced at light load currents. This idea is referred to as Pulse frequency modulation (PFM) Mode. More about the operation of PFM mode is described in the next section.

3.2 PFM Operation

A typical architecture implementing PFM mode using a hysteretic comparator as a controller is shown below.



7. Hysteretic Comparator Based PFM Architecture

Referring to the above figure, the working of the above architecture can be described as follows. If the output goes lower than the lower threshold voltage, the comparator trips and turns on the upper MOSFET. The upper MOSFET stays on till the output hits the upper threshold voltage of the comparator. The comparator trips again and turns off the upper MOSFET and turns on the lower MOSFET after a dead time. As this is synchronous buck architecture, zero current detection is required to prevent negative

current i.e. current from the load to the switches. The ripple in this architecture is decided by the hysteresis band.

While the hysteretic PFM is simple to implement, as only a hysteretic comparator is needed, its efficiency at extremely light loads (in the order of μA's to 10's of mA's) is reduced, as there is no limit put on the peak inductor current.

Though PFM control achieves higher efficiency at light load currents, the output voltage suffers from high output ripple and increased harmonics due to the variation of the switching frequency with load current. Typically, PFM mode is used when the load currents drop below 500mA approximately as mentioned in [10]. A variety of methods to detect the condition of load current falling below a threshold to enable PFM mode are described in the coming section.

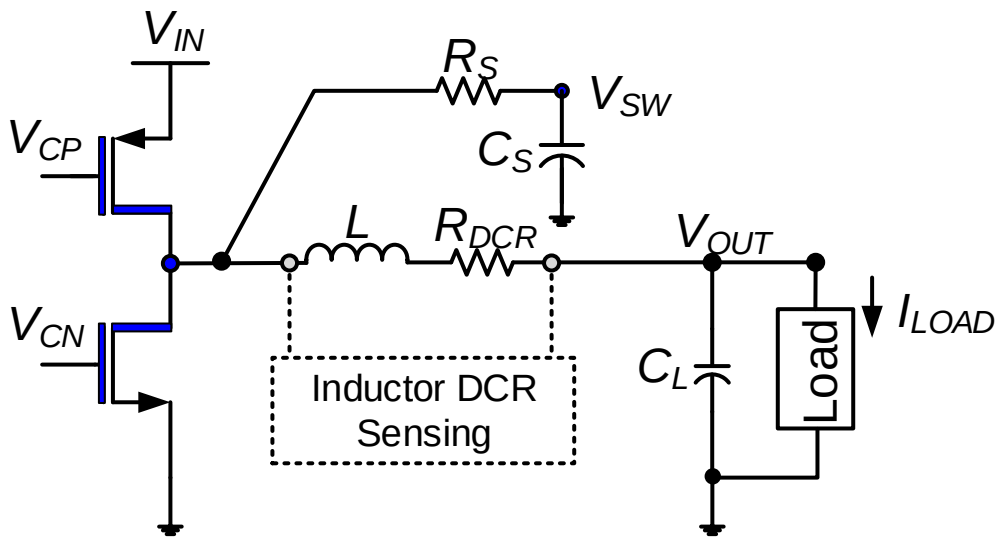
3.3 PFM Detection

The PFM detection is one of the focuses of this thesis. Some of the existing detection methods and finally, the proposed detection method are explained below.

3.3.1 Average Current Sensing Method

In this method described in [4], the idea is to sense the average current flowing through the inductor. This average current is nothing but the load current. The average current to be sensed is proportional to the DC voltage drop between switching node (V_{SW}) and output node (V_{OUT}) which is in turn equal to the DC voltage drop across the R_{DCR}. It should be noted that in steady state the inductor acts as a short and hence no DC voltage drop across it. Thus, the DC voltage drop across the R_{DCR} is a measure of load current. To obtain DC voltage component from the switching node, an RC filter is hooked up to it

and the voltage across C serves as the averaged switching node voltage (V_{SW}) whereas averaged output voltage (V_{OUT}) is already at hand except for small ripple on it which can be neglected. Figuratively, this procedure is shown below (only part of the Buck Converter is shown below).



8. Average Current Sensing Architecture

This method offers very good accuracy in sensing load current for high R_{DCR} values. But, there will be efficiency degradation due to the loss of power in R_{DCR}. Thus, an optimal value of R_{DCR} needs to be chosen based on the desired accuracy and efficiency.

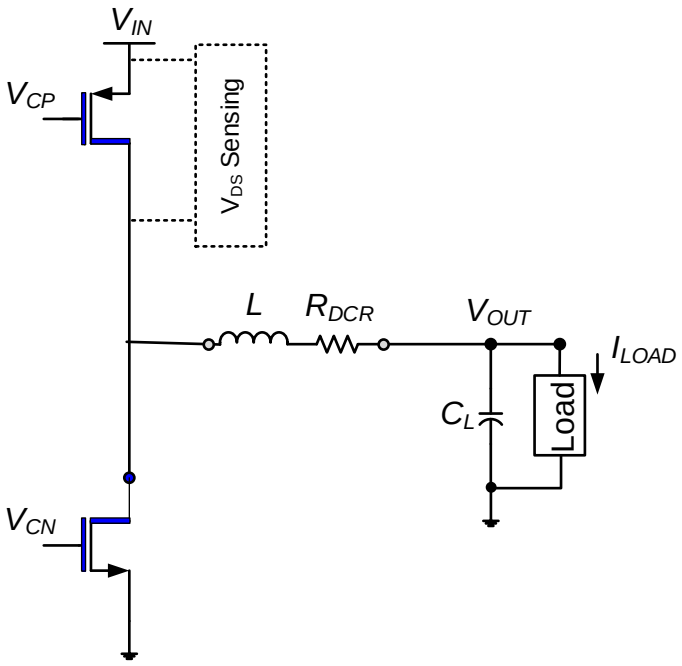
3.3.2 R_{DS} Based Current Sensing Method

In this method described in [1], the property of MOSFET acting as resistor (R_{DS}) when they are on and biased in ohmic region is used to estimate the load current. The load current is estimated by sensing the voltage across the drain-source of the MOSFET (V_{DS})

which is proportional to load current assuming the R_{DS} of the MOSFET is known. The equivalent resistance of MOSFET is given by the following equation.

$$R_{DS} = \frac{L}{W\mu C_{ox}(V_{GS} - V_T)}$$

Where μ is the mobility, C_{ox} is the oxide capacitance per unit area, and V_T is the threshold voltage. It can be noticed from the above equation that R_{DS} has significant process variations due to process dependent parameters μC_{ox} and V_T as well as temperature variations. So, this method offers poor accuracy when it comes to sensing load currents. However, it does not degrade efficiency since we are using the already existing MOSFET resistance for detection. The above described idea is summarized in the figure below.



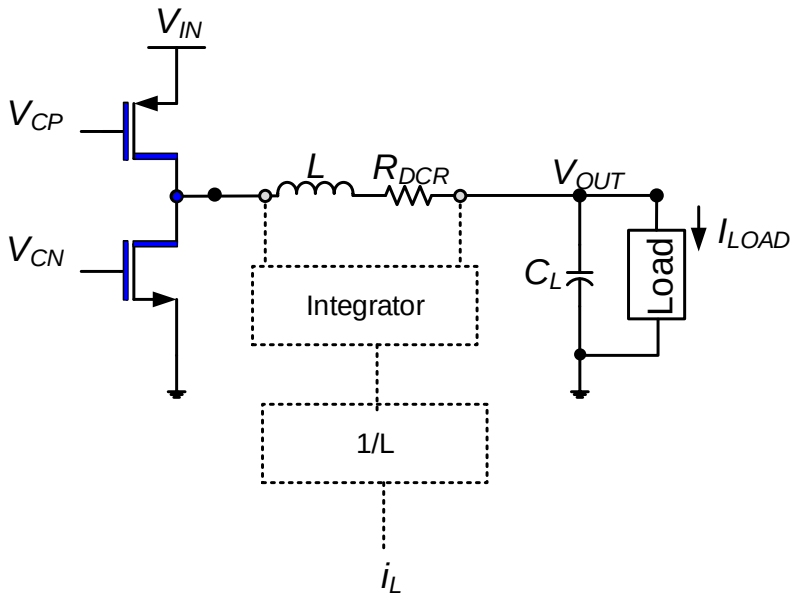
9. R_{DS} Based Current Sensing Architecture

3.3.3 Sensor less (Observer) Current Sensing Method

In this method described in [11], the voltage across the inductor is integrated and divided by inductor value to obtain the current through inductor which will have both DC and AC current. This is described in the equation below.

$$i_L = \frac{1}{L}\int v_L dt$$

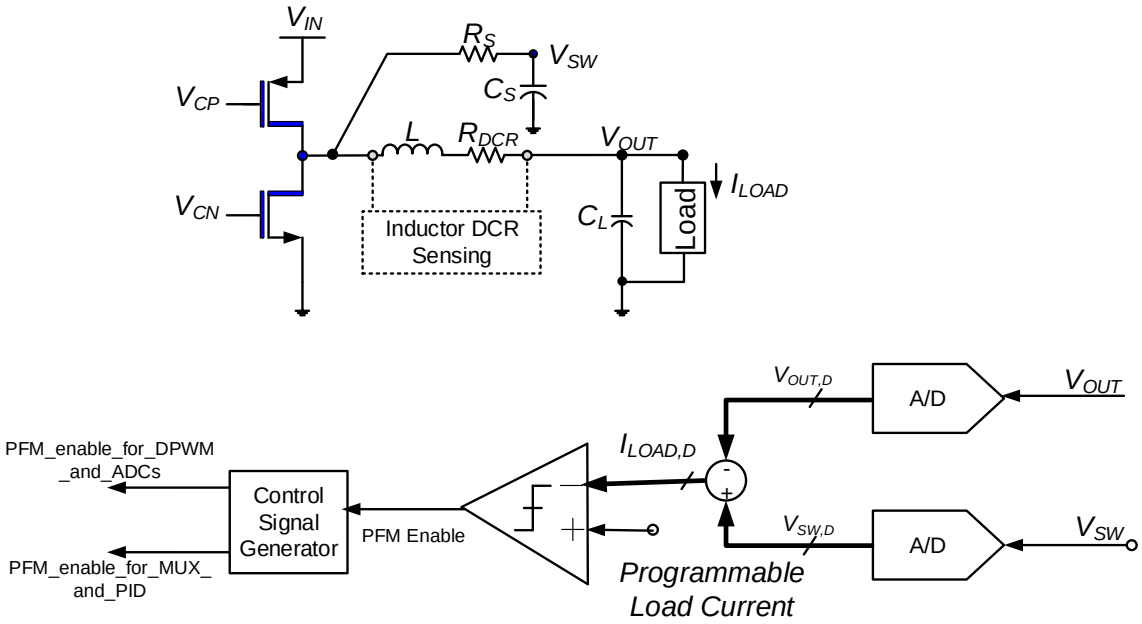
The DC current component in i_L is nothing but the load current. This method suffers from poor accuracy because it relies on the absolute value of inductor. From an efficiency point of view, this method is advantageous because other than the auxiliary circuits (which are necessary in every sensing method) like OpAmp integrator and scalar, no power is burnt for the sake of sensing load current. Pictorially, this method is shown below.



10. Sensor-less (Observer) Current Sensing Architecture

3.3.4 Proposed Digital Detection Method

From the various methods described above, it can be understood that each method trades off accuracy in sensing load current with efficiency of the Switching Converter. For this project, it is required that 'reasonable' accuracy in sensing load current as well as 'reasonable' efficiency are to be achieved. So, Average Current Sensing method is chosen so that empirically, on test board, a R_{DCR} can be found to achieve 'reasonable' efficiency and accuracy in current sensing. Also, it is required that the load current at which PFM needs to be enabled to be programmable. Hence, a digital PFM Detector is implemented whose architecture is shown below.

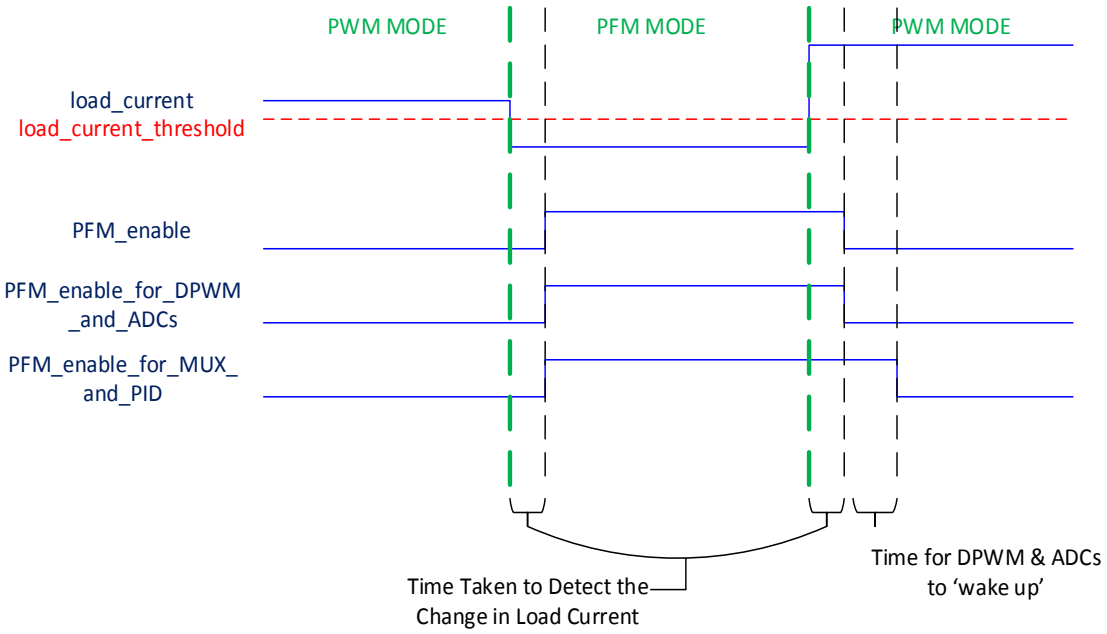


11. Digital PFM Detector Architecture

Referring to the above figure, A/D quantizes the averaged switching node voltage (V_{SW}) from which the digitized output voltage (V_{OUT}) is subtracted to obtain DC voltage drop (V_{DCR}) across the R_{DCR}. The A/D designed is a digital intensive ADC in which VCO is

the only analog circuit and the rest is all digital logic. Since V_{DCR} is a measure of load current (I_{LOAD}), it is compared with a programmable threshold value corresponding to a desired load current at which PFM mode should be engaged. The comparison is done for a time of 500µs which is larger than the transient settling time which is 100µs. This is required in order to avoid false detections due to the load transients. Finally, if the result of comparison is true, PFM mode is engaged and the Control Signal Generator generates enable signals for other blocks in the chip.

Once the chip is in PFM and the load current goes above the PFM threshold load current, PWM Detector in the chip detects this condition. Using this information from PWM Detector, Control Signal Generator generates enable signals with some timing shown below for other blocks in the chip to enable smooth PFM to PWM transition.



12. PWM <--> PFM Transition Control Signals

The timing diagram above shows the generated control signals when the chip transitions from PWM to PFM. Referring to the below diagram, shown below are the sequence of steps that occur during PWM to PFM transition.

1. When the load current drops below the load_current_threshold – a programmable digital value, PFM mode is enabled, as shown by the PFM_enable signal.
2. PFM_enable_for_DPWM_and_ADCs signal turns off the DPWM and ADCs.
3. PFM_enable_for_MUX_and_PID signal turns off the PID and allows the Hysteresis Comparator to generate the duty cycle signal for FETs.

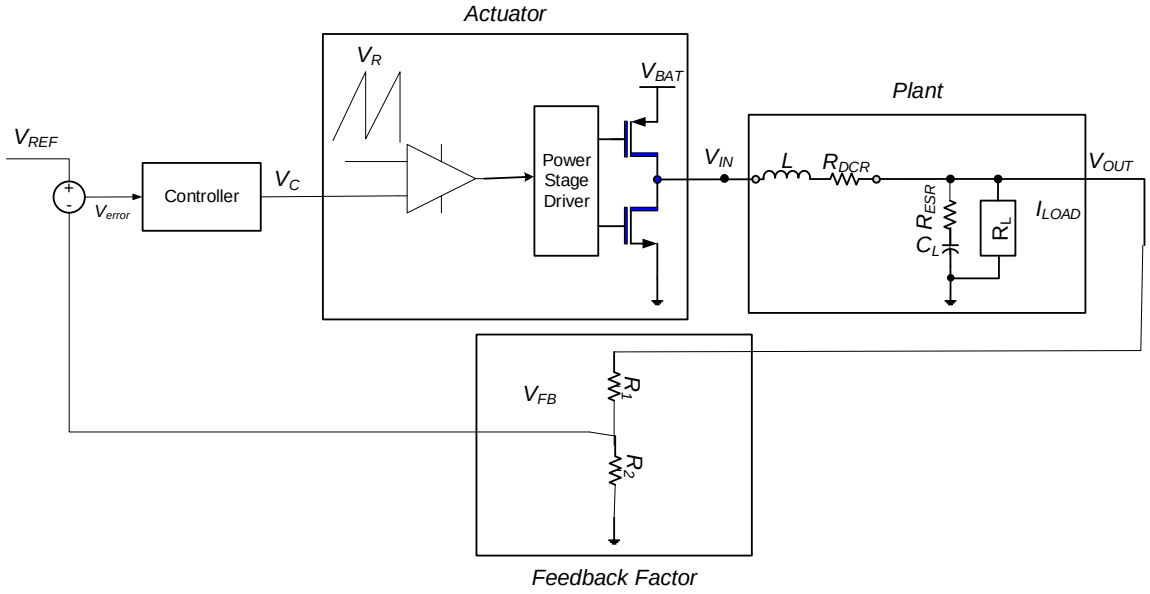
PFM to PWM Mode:

1. When the load current reaches above the load_current_threshold, PWM mode is enabled by pulling the PFM_enable signal low.
2. Some time (programmable) is allowed for DPWM and ADCs to ‘wake up’ by pulling the PFM_enable_for_DPWM_and_ADCs signal low before we toggle the PFM_enable_for_MUX_and_PID signal.
3. Finally, PFM_enable_for_MUX_and_PID is pulled low to engage DPWM, PID and ADCs to aid in generating the output voltage.

4 Modeling and Design

4.1 Modeling in MATLAB

Shown below is a typical Buck Converter. Each of the encircled boxes is named according to the standard feedback system terminology.



13. Buck Converter Feedback System

Referring to above diagram, mathematical models are derived for each of the blocks in the system.

Plant:

The pulse width modulated signal from Actuator is filtered by the LC stage yielding a DC output with 'some' ripple. Using KCL, transfer function is obtained as

$$\frac{V_{OUT}}{V_{IN}} = \frac{R_L}{R_L + R_{DCR}} * \frac{1 + sR_{ESR}C}{1 + s\left(R_{ESR}C + \frac{R_L R_{DCR}}{R_L + R_{DCR}}C + \frac{L}{R_L + R_{DCR}}\right) + s^2 \frac{LC(R_L + R_{ESR})}{R_L + R_{DCR}}}$$

It is modeled in Matlab using the continuous time transfer function block.

Actuator:

It is comprised of two blocks namely pulse width modulator and Power FET train. The pulse width modulator generates a pulse width modulated (PWM) signal as per the control voltage from the controller. This PWM signal drives the power FETs resulting in the PWM signal multiplied by V_{BAT} at the switching node. Over all the transfer function can be written as follows.

$$\frac{V_{IN}}{V_C} = \frac{V_{BAT}}{V_R}$$

In Matlab, Pulse Width Modulated signal is generated by comparing V_C against a free running ramp signal at switching frequency.

Controller:

The controller operates on the error signal which is the difference between V_{REF} and V_{FB}. The controller is modeled by means of a transfer function whose value is derived in section 3.2. In Matlab, it is modeled using the continuous time transfer function block.

Feedback Factor:

The resistor divider scales the V_{OUT} by a factor to produce V_{FB}. The relation is shown below.

$$\frac{V_{FB}}{V_{OUT}} = \frac{R_2}{R_1 + R_2}$$

The inverse of the feedback factor is approximately the gain of the feedback system. It is modeled in Matlab using a scaling block.

Now, using the models presented above, controller is designed in the next section.

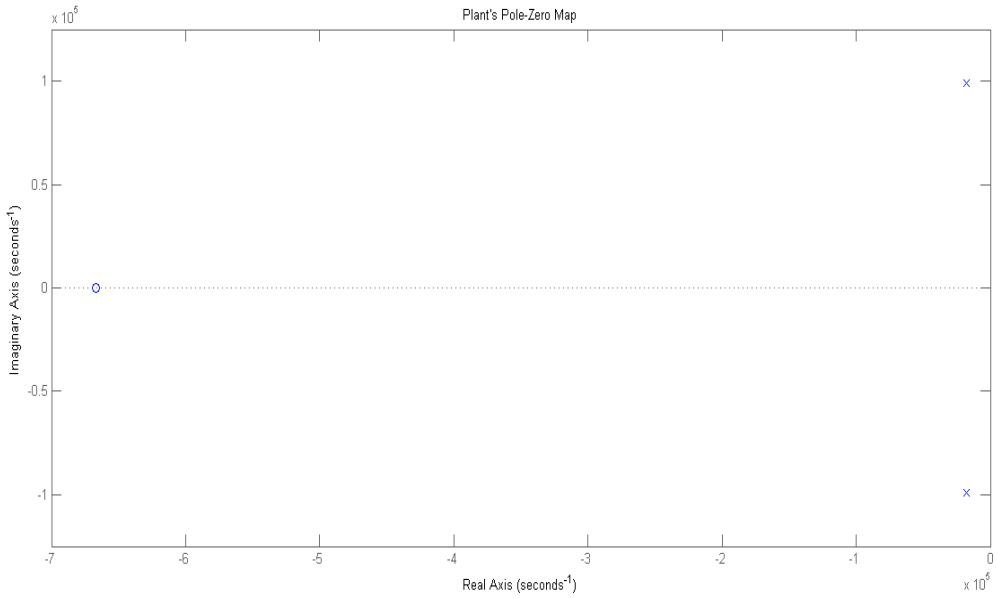
4.2 Controller Design

The controller is designed based on the chip specifications shown below.

System Specification for Buck Converter	
Input Voltage(V_{in})	3.3V-5V(5V typical)
Output Voltage(V_{out})	0.9V
Switching Frequency(f_s)	1MHz
Output Current(I_{load})	0.1A-10A
Output Voltage Ripple(ΔV)	1%-2% of V_{out}
Power Stage parameters	
L&C	
Inductor(L)	1uH-10uH
DCR(R)	50mΩ
Capacitor(C_L)	1uF-30uF
ESR	<=80m Ω
Dynamic Response	
Settling time with 0.2A(20%) load change	100us

2. Chip Specifications

Plugging in above values into the Plant model yields the following pole zero map.



14. Plant Pole-Zero Map

As can be seen in the above figure, Plant has two complex poles and one real zero. The idea is to design a controller whose poles and zeros are chosen such that a closed loop second order system with desired PM is formed. Shown below are the steps followed.

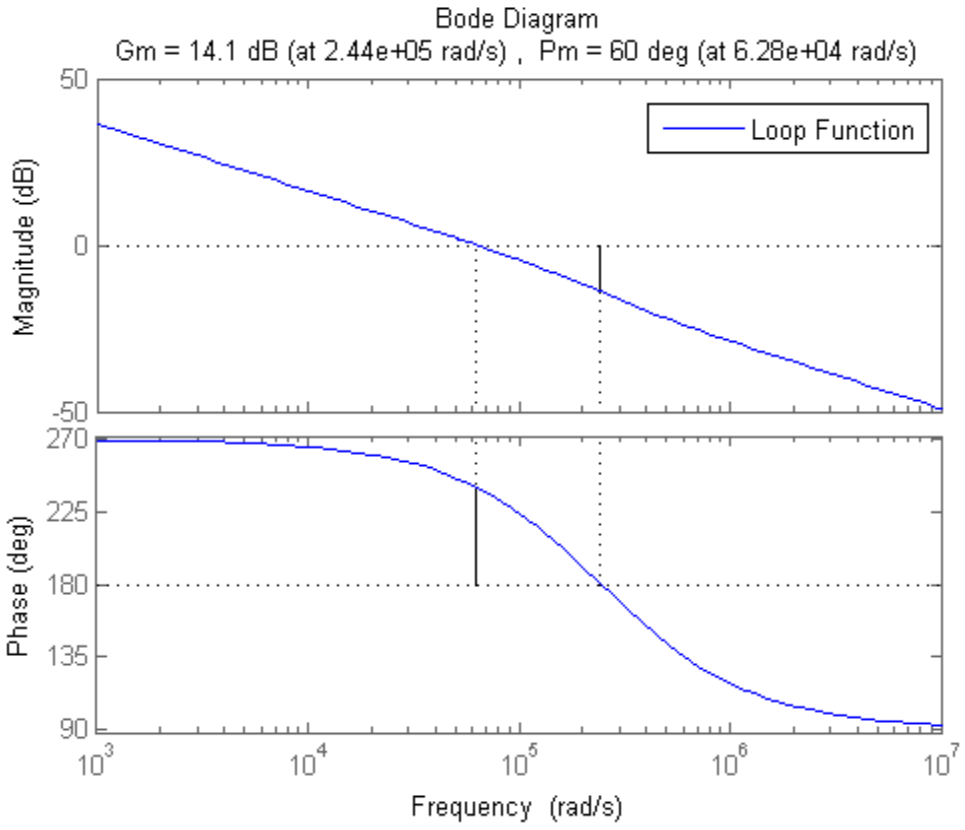
1. Two zeros are placed to cancel the two plants poles. It should be noted that the design is being done for a chosen load. When the load changes, Plants' poles changes and the controller zeros won't exactly cancel the Plants' poles. But it was observed that this has minor impact on dynamic response.
2. A pole is placed in the controller to cancel the Plant's zero due to capacitor's ESR.
3. The loop delay because of digital blocks that are going to be introduced later is modelled using Pade approximation which results in one pole and one zero. The pole is cancelled by placing a zero in controller.
4. A pole at origin is placed in the controller to have zero steady state error.
5. Depending on the UGB and PM, the controller's final poles' location is computed.
6. The loop function is constructed and scaled in order to make the gain as 0dB at UGB.

Following shown is a snippet of the Matlab code that implements the above steps.

```
%-------------------------------------------------------------------------
%  Controller Design
%-------------------------------------------------------------------------
[p z] = pzmap(plant);
% place the controller zeros to cancel the complex poles
cont_z1 = p(1);
cont_z2 = p(2);
cont_p3 = z(1);
% place controller zero to cancel the ldly_func pole
cont_z3 = ldlyp(1);
% phase due to the plant
ph_plant = atan(ugb/-z(1)) - atan((ugb-imag(p(1)))/-real(p(1))) -
atan((ugb-imag(p(2)))/-real(p(2)));
% phase due to the loop delay func
ph_ldly = atan(ugb/-ldlyz(1)) - atan(ugb/-ldlyp(1));
```

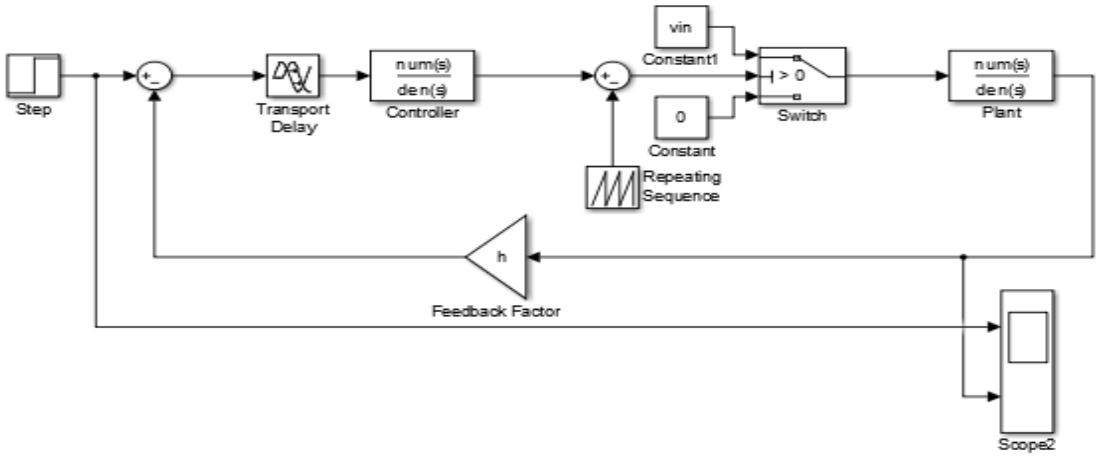
```
% phase due to the known controller poles and zeros
ph_cont = atan((ugb-imag(cont_z1))/-real(cont_z1)) + atan((ugb-
imag(cont_z2))/-real(cont_z2)) + atan(ugb/-cont_z3)...
    - (pi/2) - atan(ugb/-cont_p3);
% find where the controller's second pole needs to be placed to satisfy
phase margin
% note that controller has already pole at 0
cont_p2 = -ugb/tan(pi + ph_plant + ph_ldly + ph_cont - phase_margin);
% construct the controller without gain factor
cont_no_gain_factor = (1 - (s/cont_z1)) * (1 - (s/cont_z2)) * (1-
(s/cont_z3)) / (s * (1 - (s/cont_p3)) * (1-(s/cont_p2)));
% construct the loop func without gain factor
loop_no_gain_factor = plant * pwm * ldly_func * cont_no_gain_factor;
% make the gain of loop func 0dB at ugb
cont = cont_no_gain_factor*(1/abs(freqresp(loop_no_gain_factor,ugb)));
```

The final loop function's Bode plot obtained is shown below.

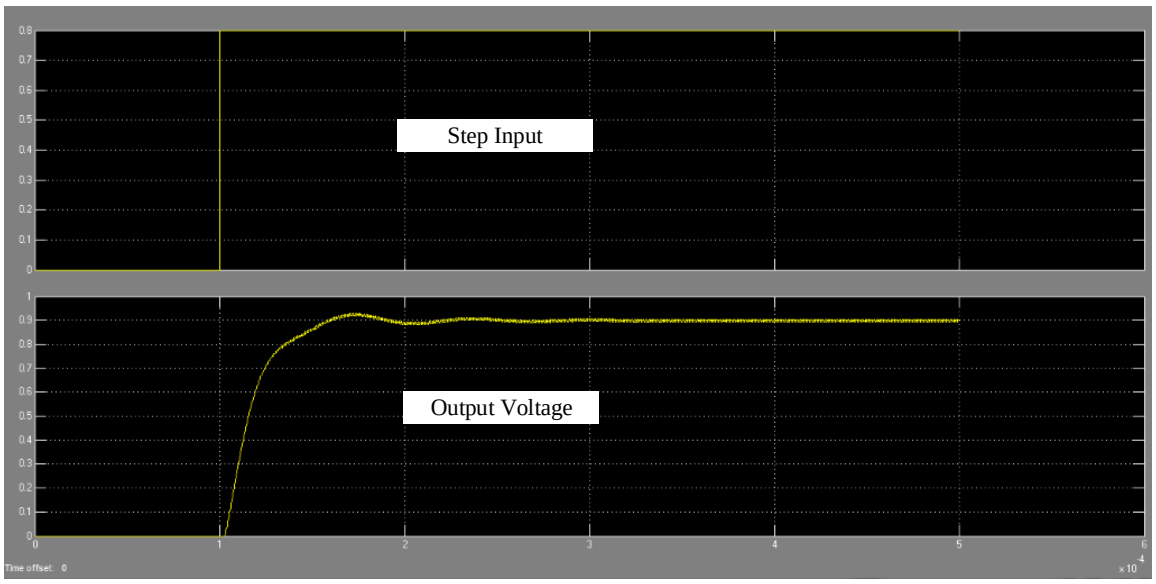


15. Loop Function Bode Plot

Using the above models and controller, Buck converter is simulated in Matlab as shown below.



16. Buck Converter with Analog Controller Model



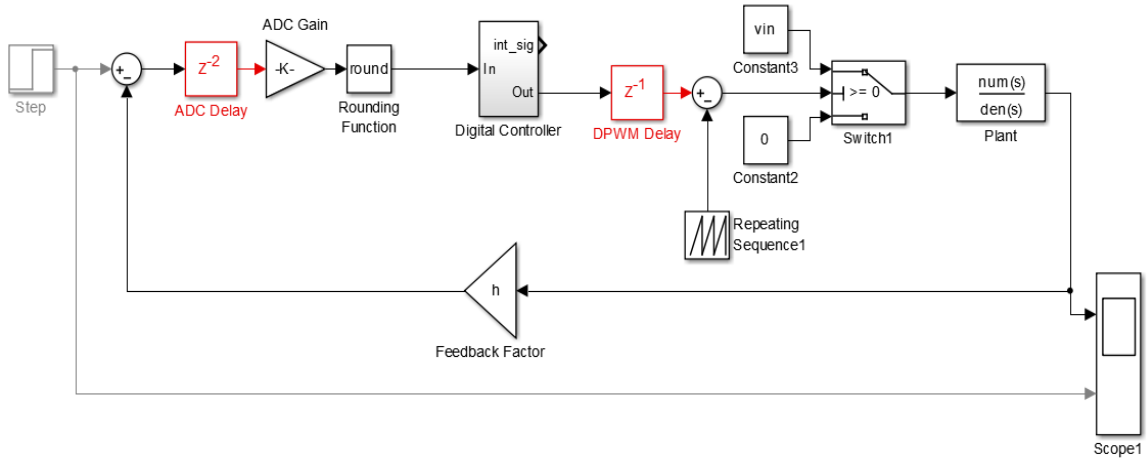
17. Buck Converter with Analog Controller Simulation

Once the continuous time Controller is designed, it is taken through various steps shown below to finally obtain the digitized controller.

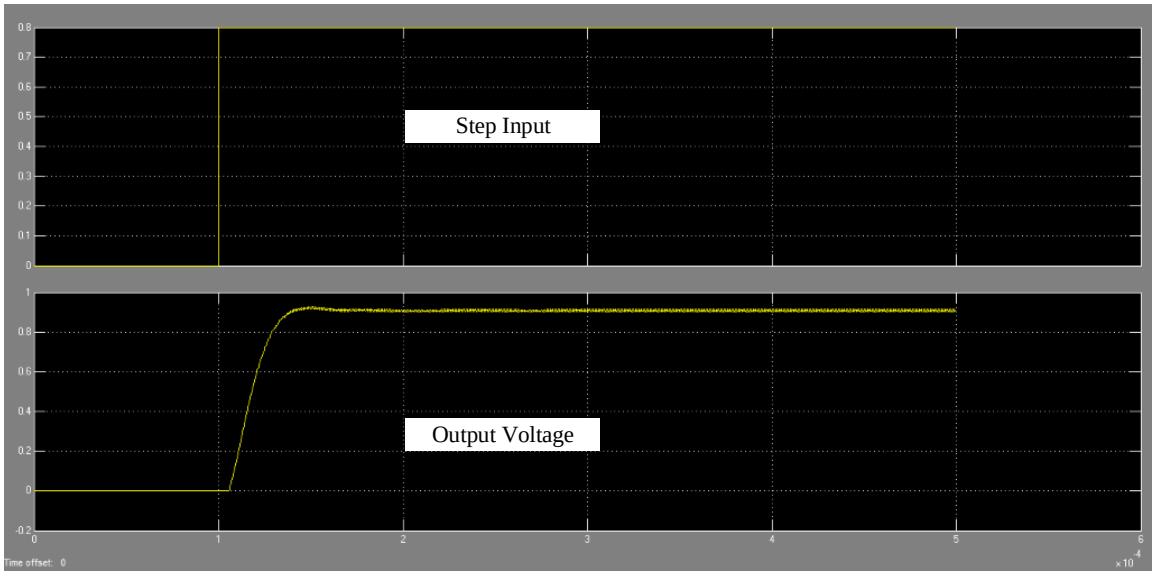
1. The continuous time Controller is discretized using Bi-linear transform. Then, the discretized transfer function is digitized by scaling and rounding the numerator and denominator coefficients appropriately.
2. The next step is to introduce ADCs and DPWM to see the effects of quantization noise. Though ADC resolution designed for this chip is 10 bits, it can be programmed though SPI to operate with resolution anywhere from 6 to 10 bits. This feature is added to overcome the effect of limit cycles as described in [8]. The DPWM resolution is fixed at 9 bits. The digital gain (equation shown below) added into the loop because of these two blocks must also be taken into account while designing controller.

$$ADC_{gain} * DPWM_{gain} = \frac{2^{ADC\ bits}}{ADC\ Voltage\ Range} * \frac{1}{2^{DPWM\ bits}}$$

Though the Matlab model is incrementally developed and verified at all stages of the design, for convenience, the full model along with the simulation result is shown once below.



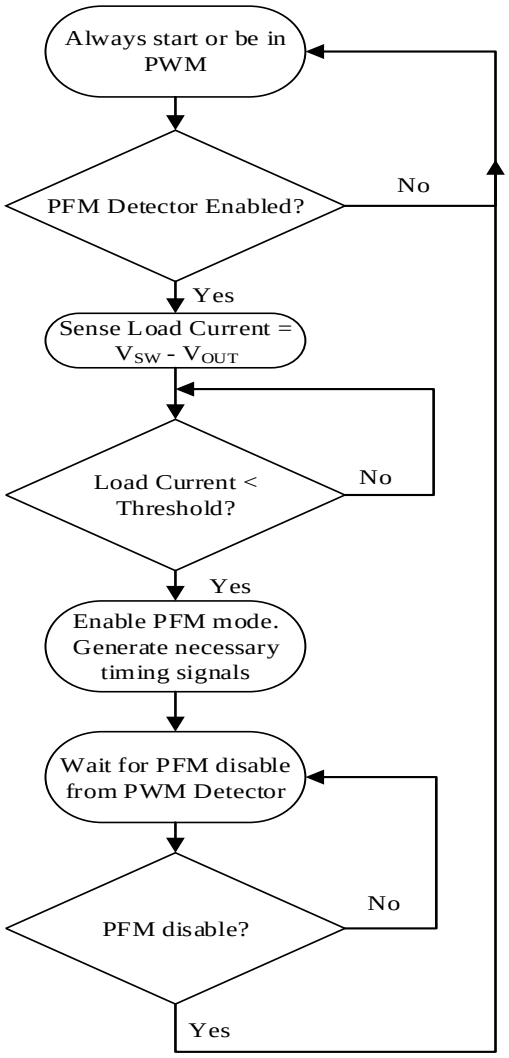
18. Buck Converter with Digital Controller Model



19. Buck Converter with Digital Controller Simulation

4.3 PFM Detector Design

As explained in the section 3.3.4, the PFM mode is detected based on the DC drop across the inductor. Even if the inductor R_{DCR} varies, the threshold to switch into PFM mode can be reconfigured through SPI. Using the values in the specification table in section 3.2, the minimum DC drop that needs to be resolved is computed to be R_{DCR} * I_{PFM}, where I_{PFM} is the load current below which PFM mode needs to be enabled. In case it is tough to resolve the above value well with the ADC, an external amplifier is used to amplify the averaged V_{SW}. Once the digitized values of V_{SW} and V_{OUT} are available, PFM is detected based on the flow chart in the next page.



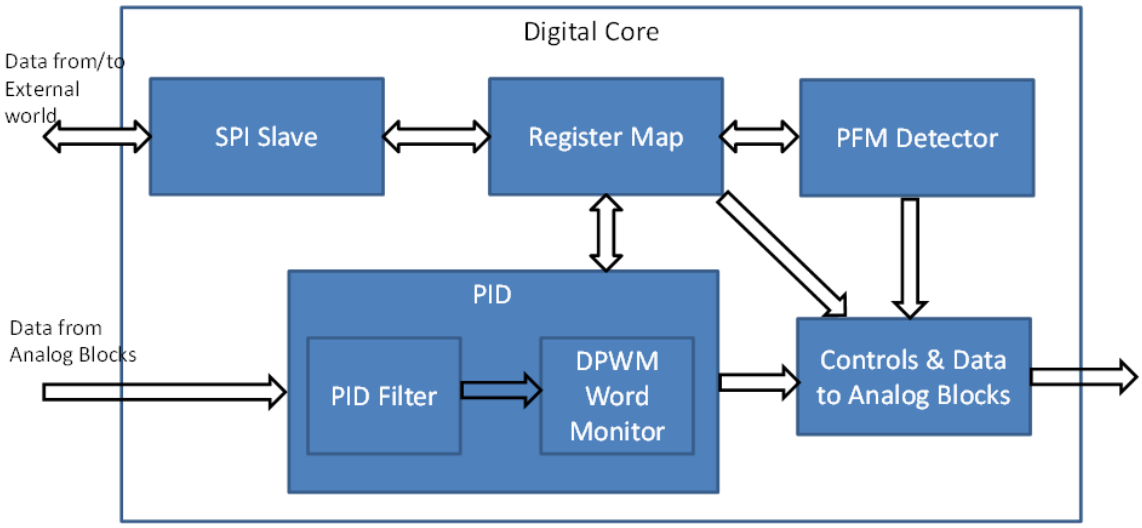
20. PFM Detection and Operation Flow Chart

5 DIGITAL DESIGN AND IMPLEMENTATION

5.1 RTL Design

5.1.1 Digital Core

The Digital Core is comprised of three main blocks from the functionality point of view. They are Digital Controller, PFM Detector, SPI and High Speed SPI. In addition, Digital Core has many supporting blocks namely Clock Generator, Reset Synchronizer, Register Map, Data Transfer. All these blocks are described in coming sections. Shown below is the Digital Core architecture.



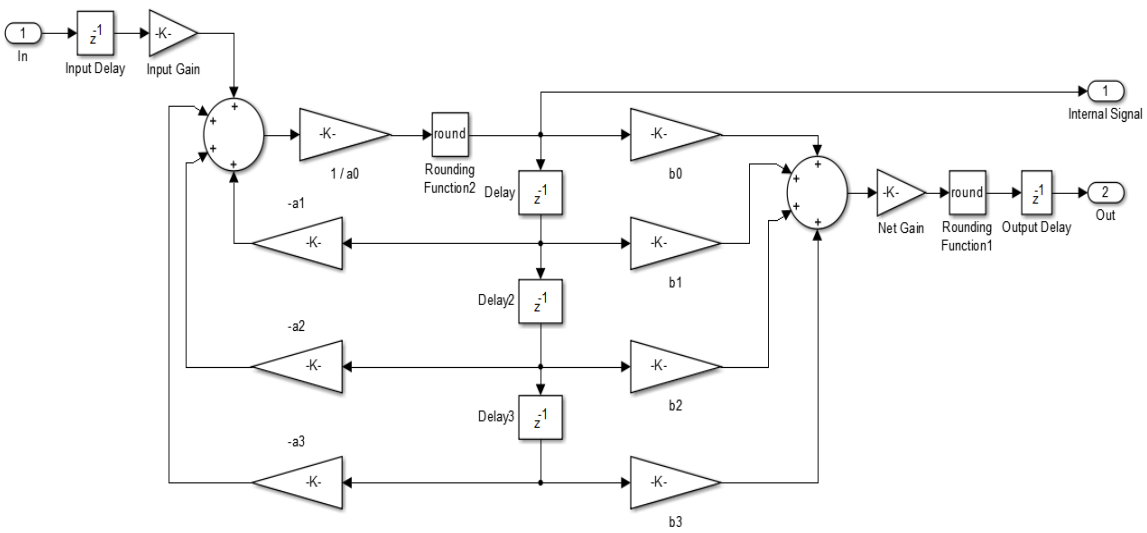
21. Digital Core Architecture

5.1.2 Digital Controller

The Digital Controller general form is as follows.

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}$$

The architecture shown below is chosen based on Matlab simulations to implement the above transfer function.



22. Digital Controller Architecture

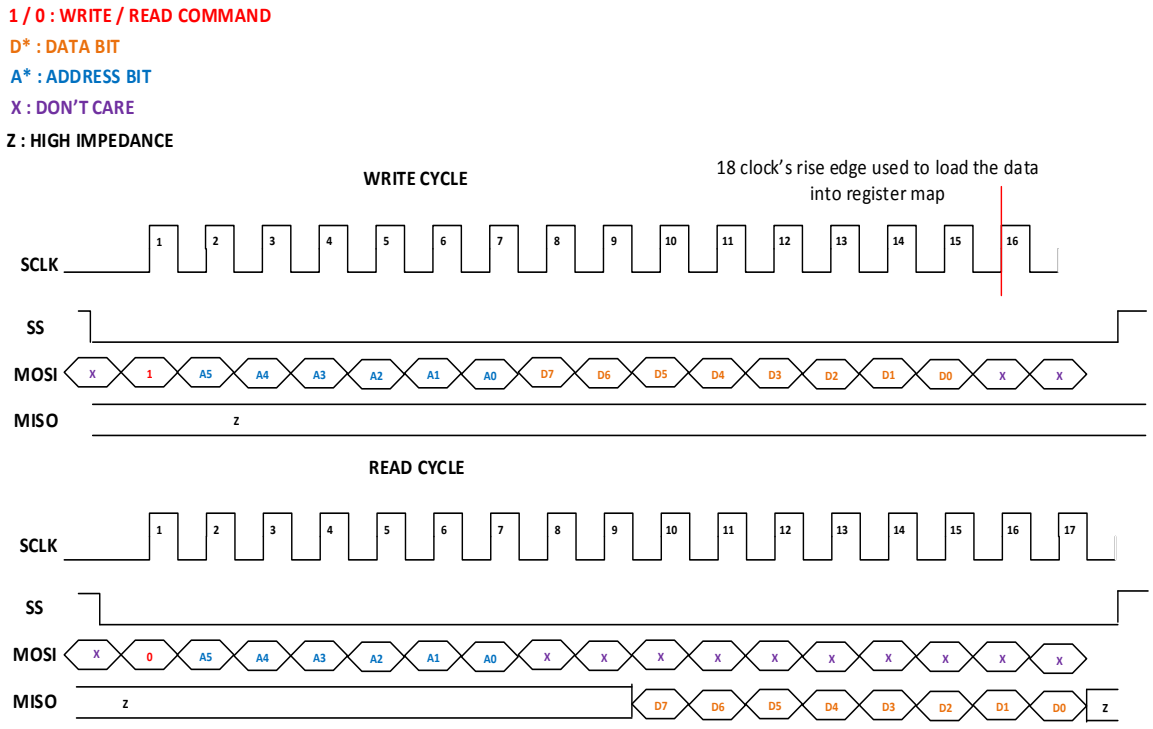
The architecture in the above figure is described using Verilog and then mapped to gates. In the above figure, each z^{-1} represents a bank of flip flops with size equal to the bit widths of signal at its output node. The widths of the signals are decided based on how high value a signal takes for 'reasonable' simulation conditions. The widths of the coefficients are chosen such that they give acceptable performance as that of their discrete counter parts. All multipliers and adders required are inferred from synopsis design ware library.

5.1.3 PFM Detector

The logic implementing the PFM Detector functionality described in section 3.3 is described in Verilog and then mapped to gates.

5.1.4 SPI

The protocol for this slow serial interface is shown in the next page.

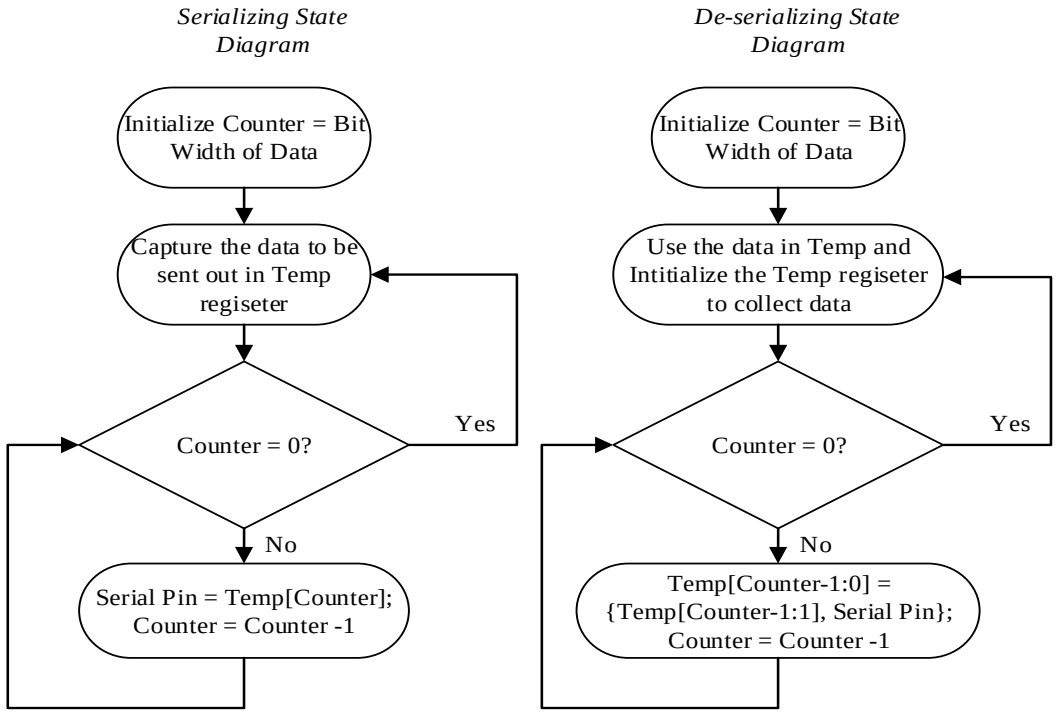


23. SPI Timing Diagram

A state machine that implements the above timing sequence is described in Verilog. This interface is used to program and enable debug modes in the chip.

5.1.5 High Speed SPI

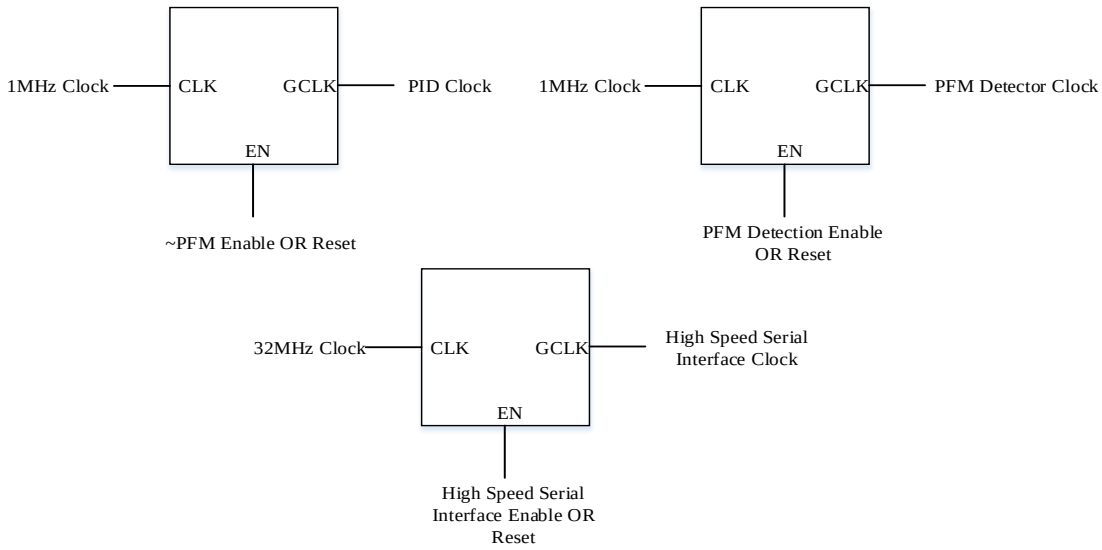
The state diagram for the high speed serial interface is shown in the next page. A state machine that implements the sequence of events in the state diagram is described in Verilog. This interface is used to ship data in or out of the chip in debug mode and also to characterize mixed signal blocks in the chip.



24. High Speed SPI State Diagram

5.1.6 Clock Generator

The Clock Generator architecture is shown below.

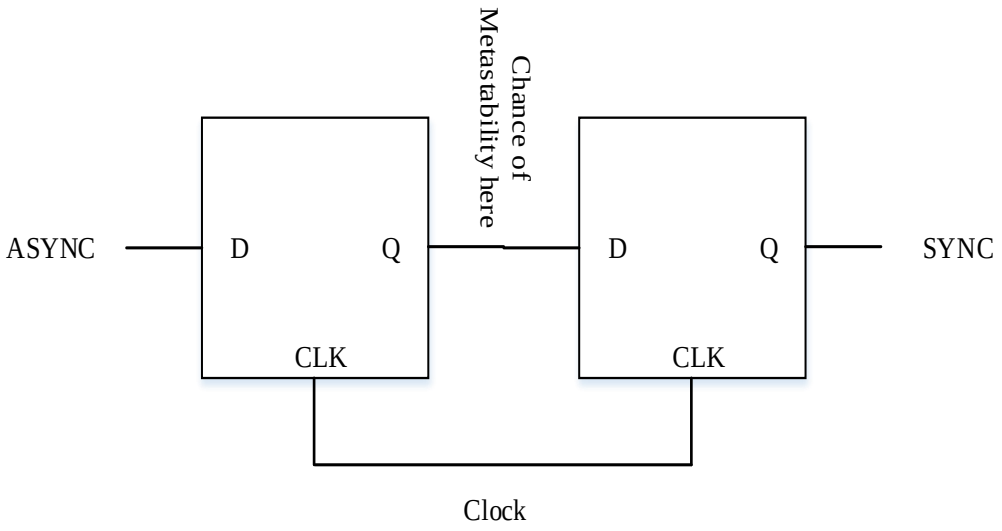


25. Clock Generator Architecture

Depending on the operating mode, clocks are generated or killed to save power. Latch based clock gating cells are used since they generate glitch free clocks.

5.1.7 Reset Synchronizer

The Reset Synchronizer is used to synchronize the asynchronous chip reset onto various clock domains inside the chip such as PID clock, PFM Detector clock, High Speed SPI etc., A very widely used double buffering technique is used for this purpose which is shown below.



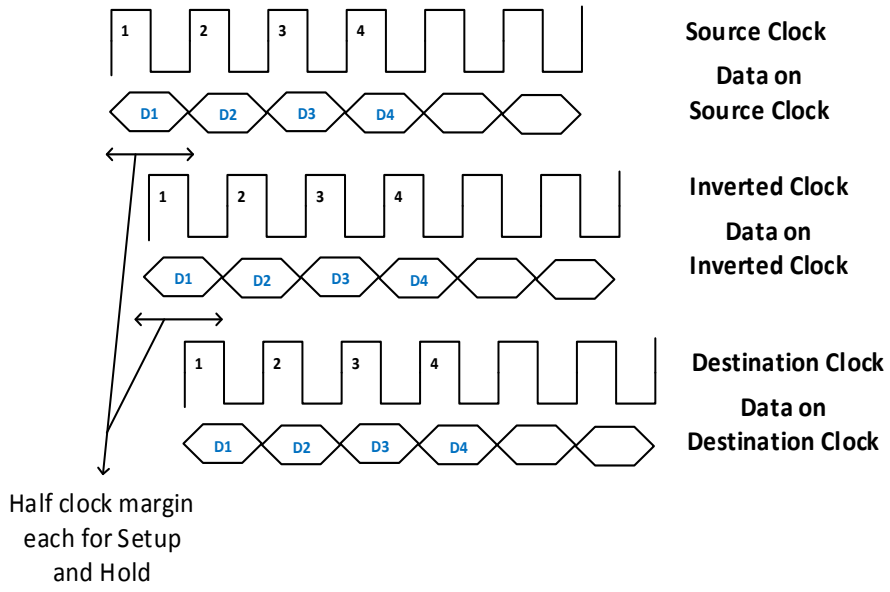
26. Reset Synchronizer

Referring to the above figure, the first flip flop captures the ASYNC (asynchronous data) and its output may become metastable. However, this metastable condition is resolved into proper logic state given enough time which is nothing but the clock period. Then, the second flip flop captures the proper logic state which is on the same clock domain. Thus, this technique synchronizes asynchronous data on to a clock domain. It should be noted

that the SYNC signal may be double or single delayed version of ASYNC depending on to what state the metastable condition resolved to. This should not be a problem for signals like Reset.

5.1.8 Data Transfer

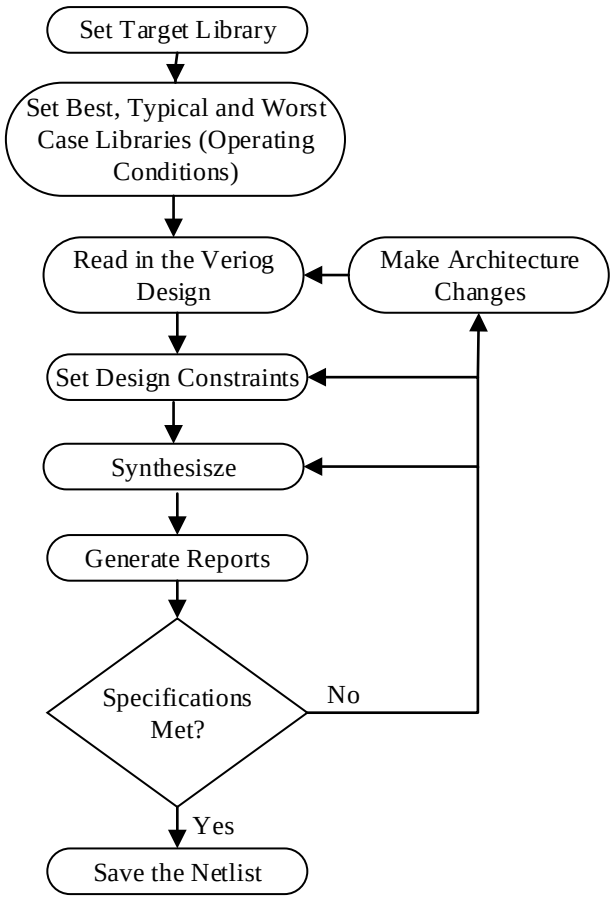
To ensure safe data transfer between different clock domains say between ADC and PID, or between PID and DPWM, the data on the source clock is transferred first onto the inverted destination clock and then onto the destination clock. This way, good setup and hold margins are ensured during data transfer. The timing diagram describing the above procedure is shown below.



27. Data Transfer between Clock Domains

5.2 Synthesis

The Synthesis methodology is depicted in the flow chart below.



28. Synthesis Methodology Flow Chart

The steps in the above flow chart are elaborated below.

About Target Library:

1. The target library used is 1.8V, 0.18µm radiation hardened IBM standard cell library. It comprises of variety of digital circuits such as combinational gates, latches, flip flops, clock buffers and clock gating cells. However, latches and asynchronous flip flops are avoided to overcome the difficulties in timing analysis.

Operating Conditions:

2. The chosen library comes with characterization data corresponding to three operating conditions, usually referred to as worst, typical and best case conditions. The nomenclature signifies the delay of the cells.

Design Constraints:

1. Any nets which are to be synthesized in APR (Auto Place and Route) stage (since APR tool will have more accurate info about delays) are set do not touch. Two such nets are Reset and all Clock nets in design.
2. The boundary conditions for the digital core are set. Basically, this information is to let the tool know about the drivers of the input pins, and the loads being driven by output pins. The drivers or loads can be categorized into two kinds of cells viz. flip flops and pads. Accordingly, boundary conditions are set for input and output pins.
3. All Clocks in design are created and their attributes are set. Attributes include Clock Latency, Clock Uncertainty and Clock Transition. Constraints are also set to do clock gating checks. Also, constraints to fix hold violations are used if the slack is too high, if not they are left unfixed to be later fixed during APR.
4. The input and output delays are used to define the relationship between data and clock before and after the digital core respectively. This information is needed to ensure that they are captured correctly as soon as they hit the core and they leave the core.

The signals which are asynchronous like reset, debug signals etc., to the digital core are ignored under this category.

5. In the digital core, there are many signal paths from SPI clock to various functional clocks like PID clock, PFM detector clock and high speed SPI clock. For practical purposes these signals can be considered pseudo static meaning they change once in a while and their moment of change is not so important. So it does not make sense to analyze the timing for these paths which is why false paths are set.
6. In the digital core, some signals are transferred from a slow clock to fast clock and vice versa which have a fixed frequency relation. One such example is when data is transferred from 1MHz clock to 32MHz clock (and the other way) for serializing (de-serializing) and shipping out (into) of the chip. In scenarios like these, the timing need not be analyzed between immediate rising edges of source and destination clocks. In order to let the tool know about the above scenarios, multi cycle paths are set in the digital core.

More information about design constraints can be found in [12] and [13] (Synopsys manuals). Finally, the results of the Synthesis are tabulated in the next page. Since the Hold-Total Negative Slack is less, it is fixed in Place and Route stage in which there is more accurate information about delays.

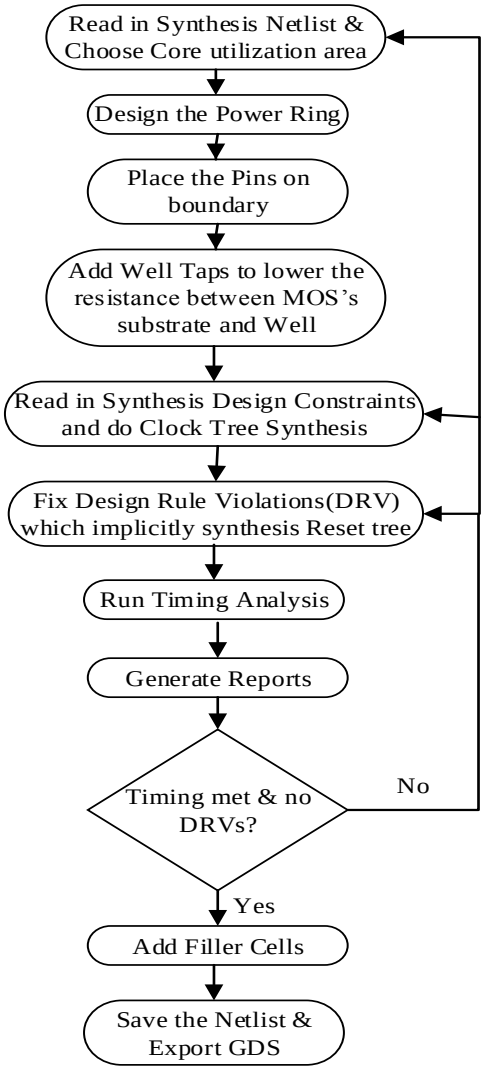
Parameter	**Value**
Area	$348008 \mu m^2$

Setup-Total Negative Slack	0ns
Hold-Total Negative Slack	6.68ns

3. Synthesis Results

5.3 Place and Route

Shown below is a flow chart depicting various steps in Place and Route.



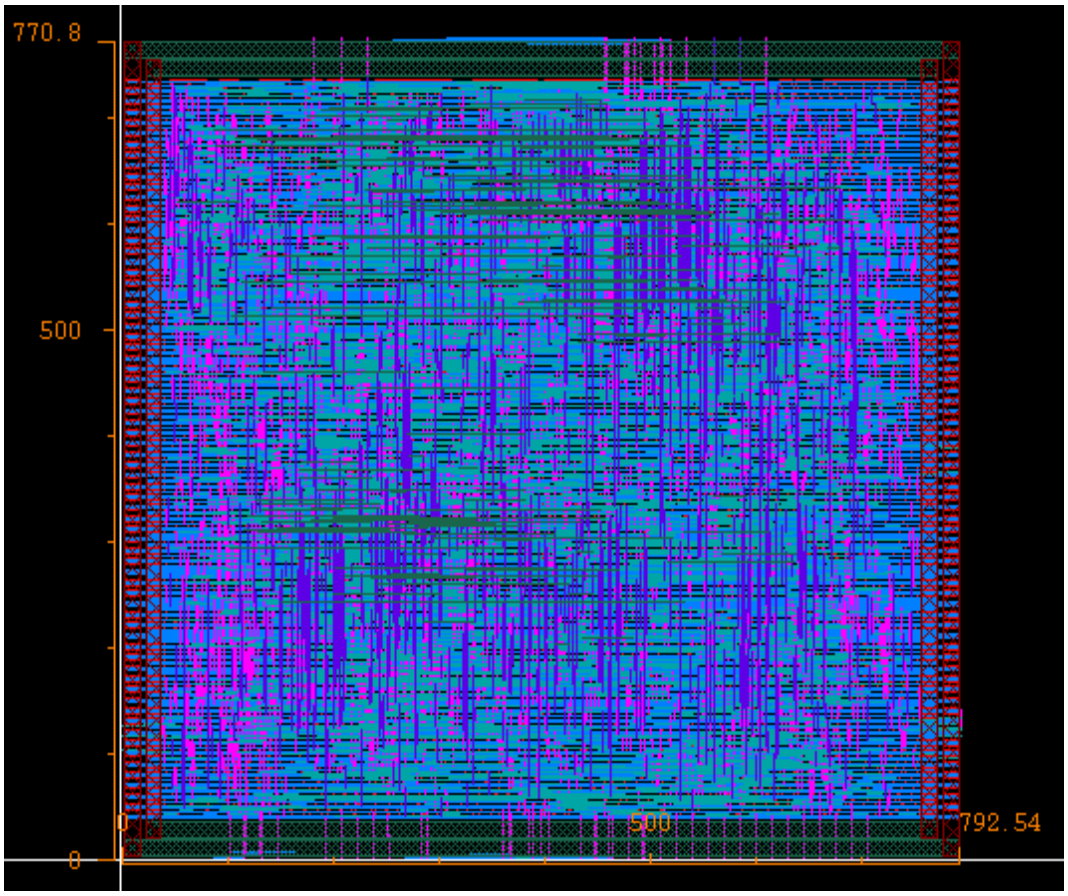
29. Auto Place and Route Flow Chart

The gates obtained during Synthesis are laid out using the Cadence SOC Encounter tool.

Some of the steps shown above are elaborated below.

1. The width of the power ring and power stripes are chosen big enough in order to minimize the voltage drop between power supply and power pin of the gate(s).
2. The pins are placed on the boundaries such that routings between digital core and other blocks is done with minimal wire lengths.
3. The well taps are added as mentioned in the IBM technology manual in order to have well connections in each row of the layout.
4. The Clock Tree Synthesis is done with similar constraints used while doing the Synthesis. The target here is to reduce the clock skew.
5. The Reset tree which is a high fan out net is synthesized after Clock Tree Synthesis so that there is no transition violation on any net connected to the Reset pin.
6. The Filler cells are added to fill the gaps in the layout which are required to maintain continuity of the well.

Finally, from the layout, gate level netlist is exported with which mixed signal chip simulations are run in Cadence Ultrasim. Also, GDS is exported for using in making top level chip layout. More information about using the tool can be found in [14] (Cadence manual). Shown in the next page is the digital core layout obtained using the above procedure.



30. Digital Core Layout

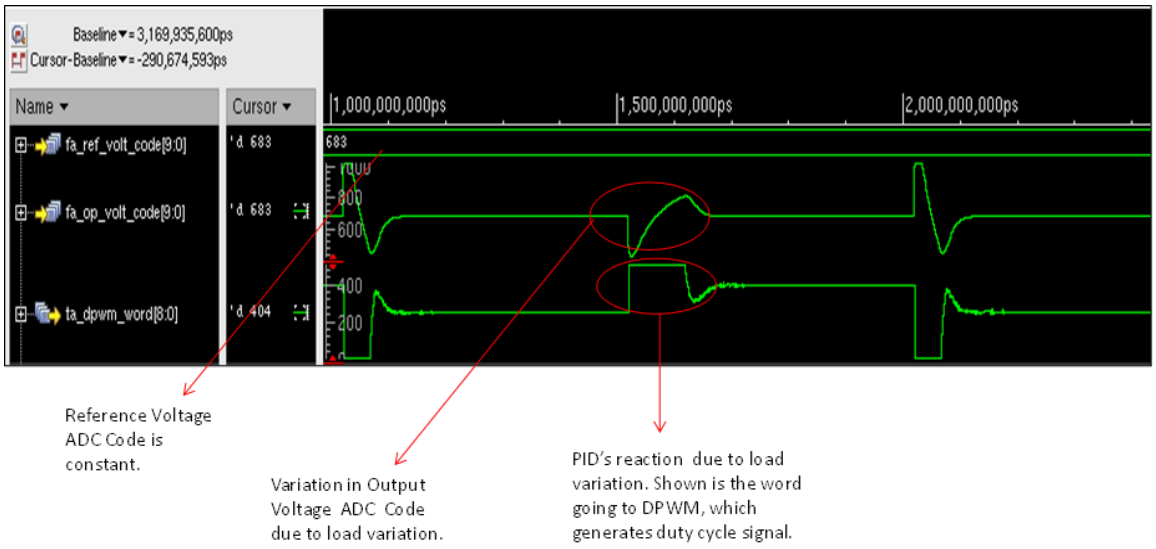
6 RESULTS

6.1 Simulation Results

In this section, the gate level simulation results are presented for each of the blocks in Digital Core.

6.1.1 Digital Controller

Shown below is a simulation result.



31. Digital Controller Simulation Result

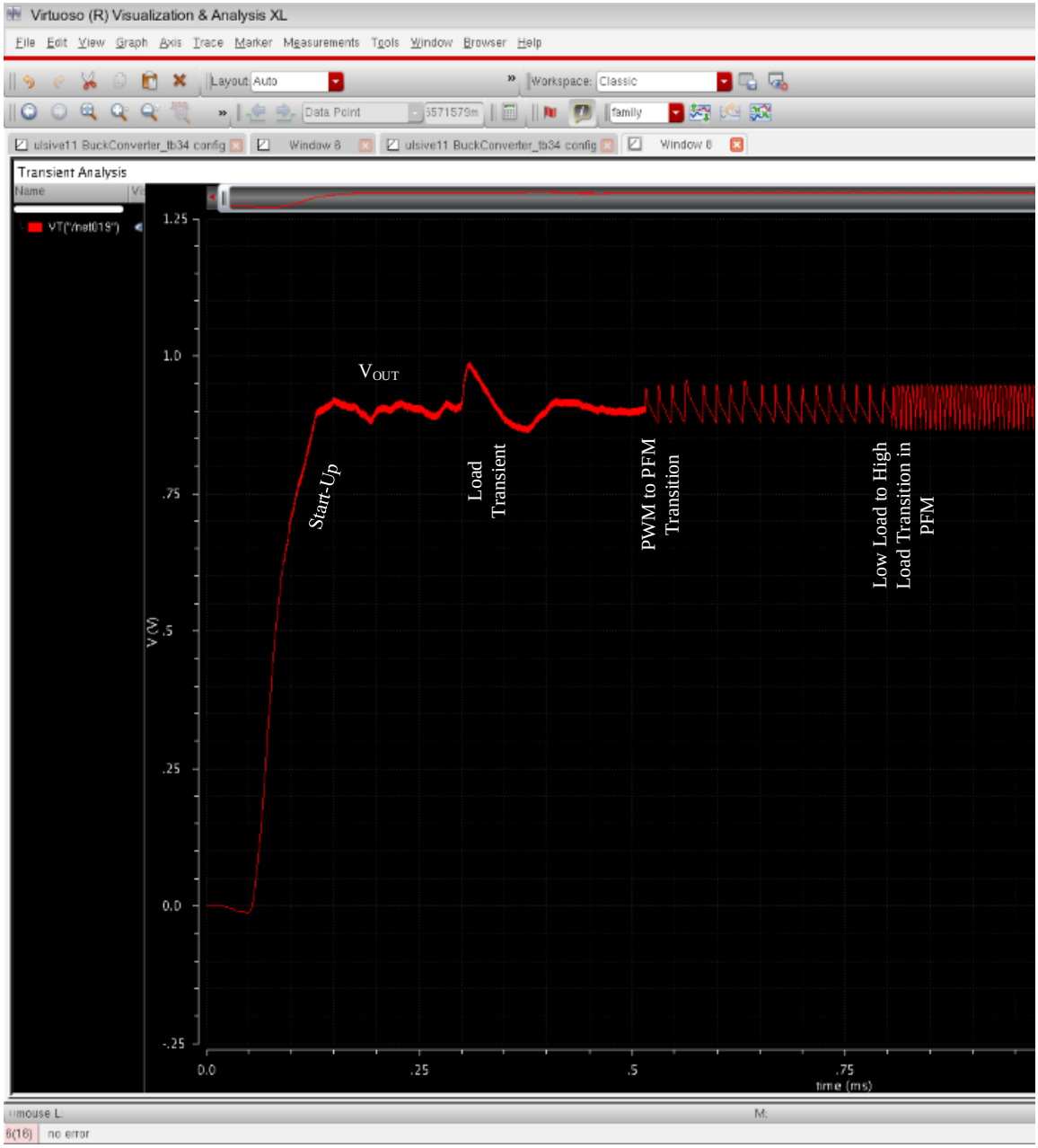
To verify the Digital Controller, firstly, simulation is run in MATLAB with some input data and the corresponding output data is stored as a reference. Then, RTL simulation is run with the same input data and the obtained output data is compared with the reference. Shown below is a simulation result.

6.1.2 PFM Detector

The PFM Detector is verified standalone by emulating switching node and output node ADCs in test bench. Codes required to enable PFM are fed from test bench and verified if the PFM Detector generates necessary control signals with expected timing between them.

6.1.3 Full Chip Simulation

Chip level simulations are run with all blocks at transistor level expect Digital Core which is abstracted to be at behavioral gate level netlist obtained after place and route. Shown in the next page is a simulation result in which, the ripple in PWM mode is about 15mV where as in the PFM mode it is around 80mV.

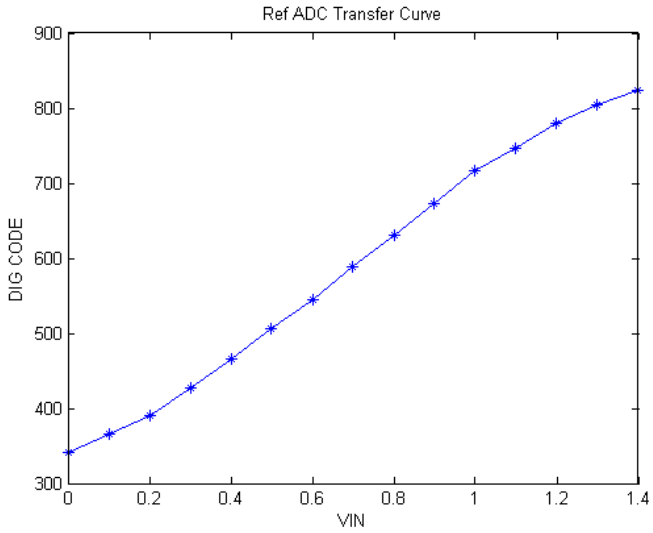


32. Full Chip Simulation Result

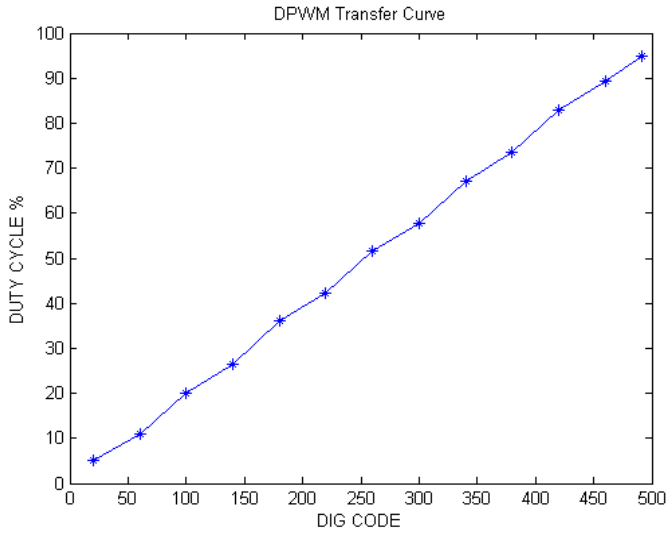
6.2 Chip Results

6.2.1 High Speed SPI

The High Speed SPI is enabled using SPI. First, this interface is successfully tested by writing different values to a SPI register and shipping it out. Then, this proven interface is used to characterize the Reference ADC and DPWM, whose results are shown below.



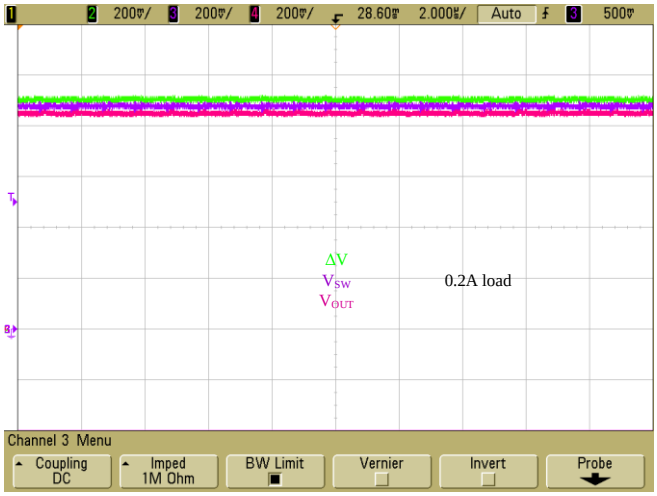
33. Reference ADC Transfer Curve Characterization



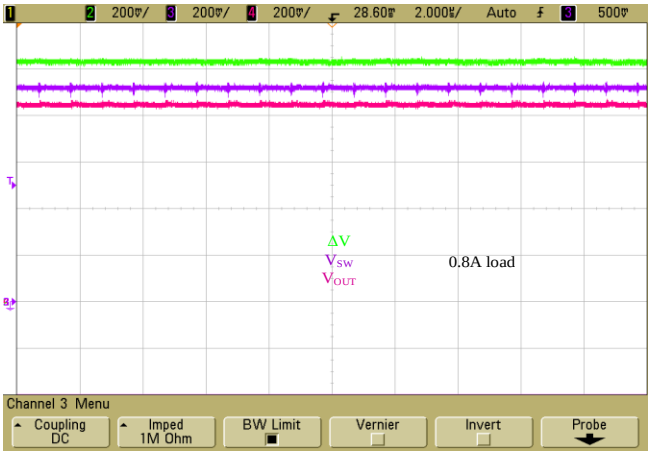
34. DPWM Transfer Curve Characterization

6.2.2 PFM Detector

Before testing the PFM Detection logic, the current sensing functionality is tested. A 50mΩ resistor is placed in series with Inductor to sense the load current which is proportional to voltage difference (ΔV) between averaged switching node voltage (V_{SW}) and output voltage (V_{OUT}). The ΔV increases with increase in load current as shown in the results below.

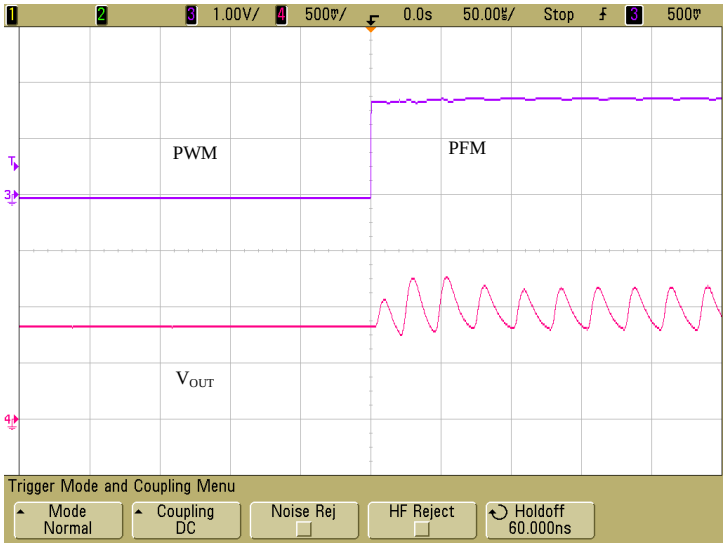


35. Current Sense with Low Load



36. Current Sense with High Load

Once current sensing is tested to be working correctly, PFM Detection test is done. The chip is brought up on a load current of 1A and PFM Detector is enabled. The load current is now decreased gradually and PFM mode kicks in at a load current of 0.5A approximately. The threshold for mode transition is determined empirically and programmed through SPI. The result is shown below.

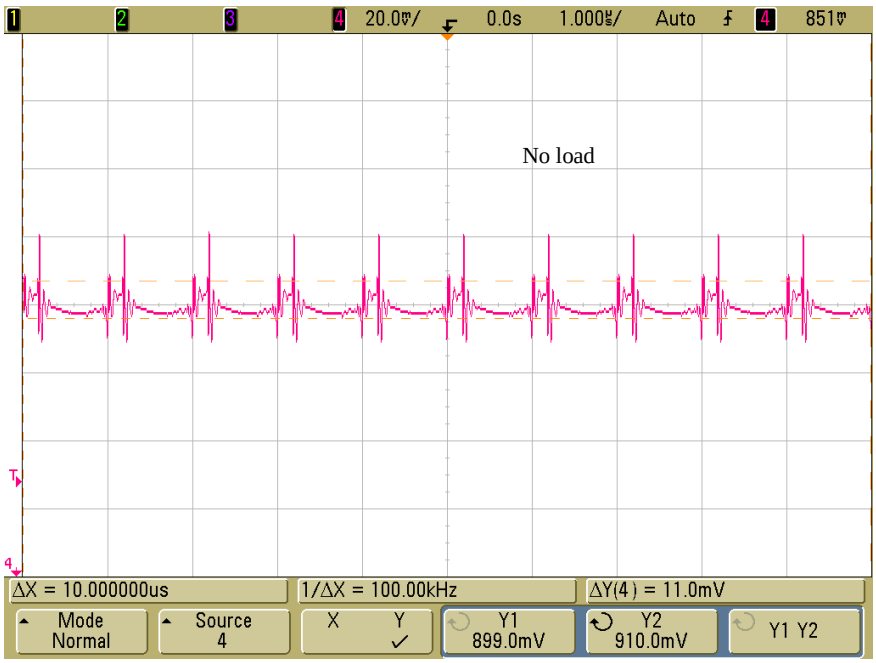


37. PWM to PFM Transition

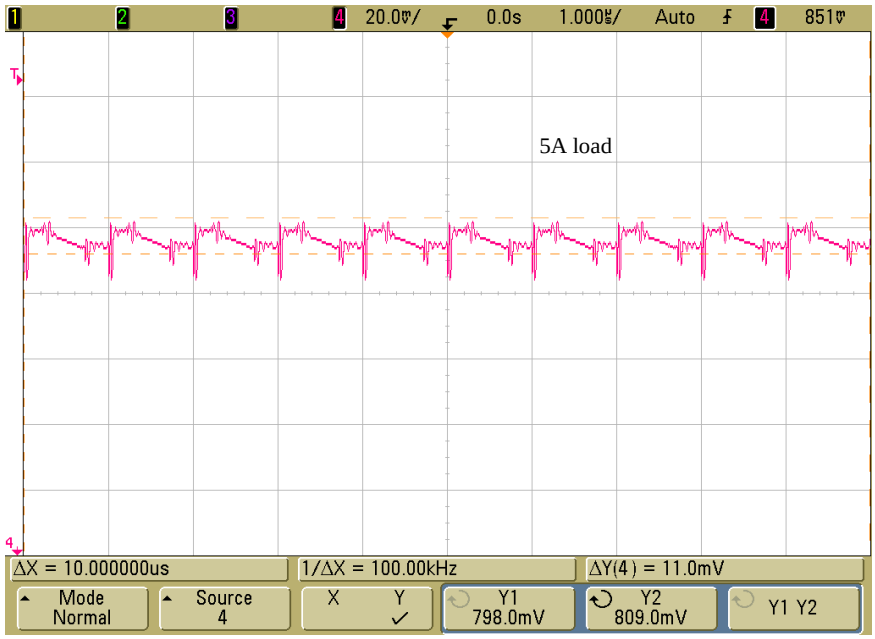
Referring to the above figure, the ripple in PFM mode is somewhat high which is to be fixed in the next revision.

6.2.3 PWM Mode – Steady State

The chip is tested in PWM mode for load currents from 0 to 5A. The results are shown in the next page.



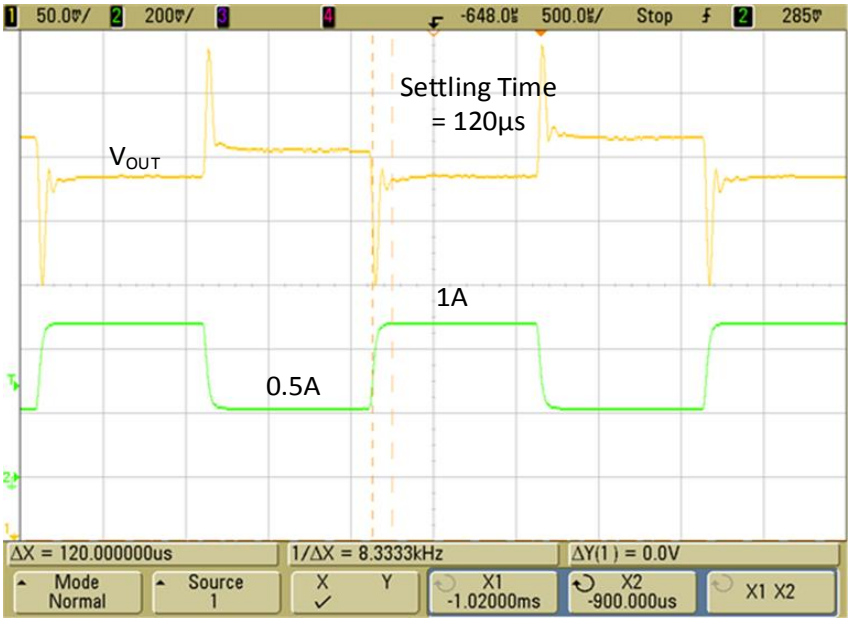
38. Output Voltage in PWM Mode with No Load



39. Output Voltage in PWM Mode with 5A Load

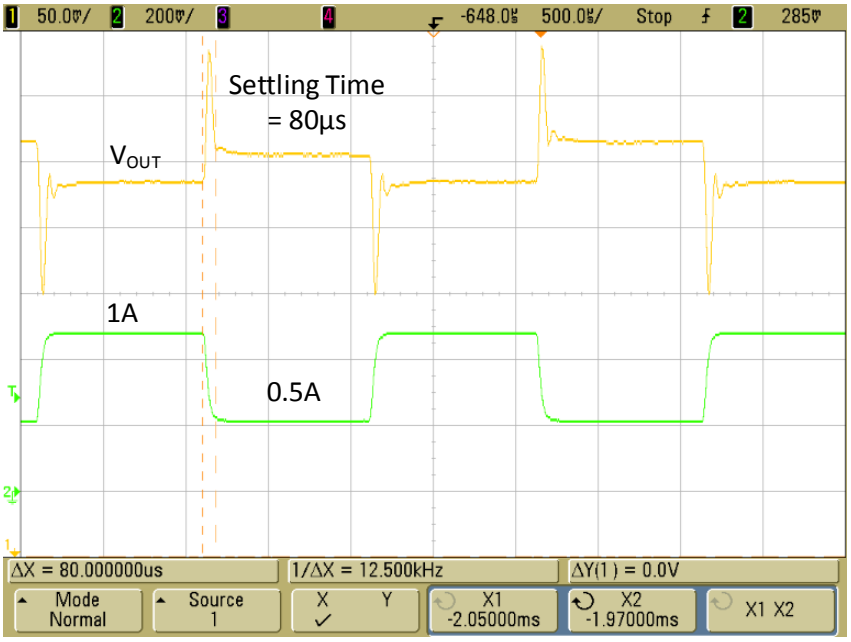
6.2.4 PWM Mode – Transient State

The load is changed from 0.5A to 1A and V_{OUT} is captured as shown below.



40. Settling Behavior for 0.5A to 1A Load

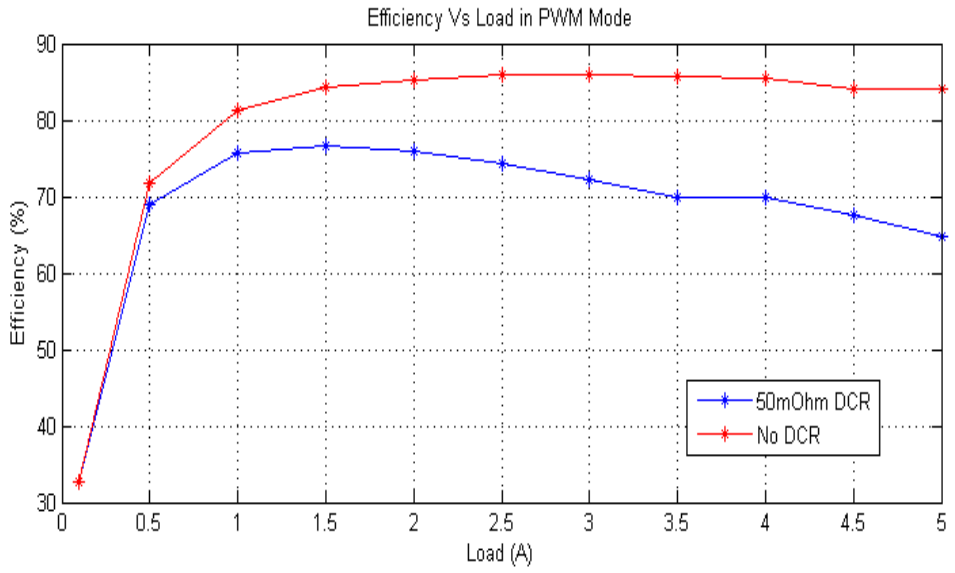
The load is changed from 1A to 0.5A and V_{OUT} is captured as shown below.



41. Settling Behavior for 1A to 0.5A Load

6.2.5 Efficiency in PWM Mode

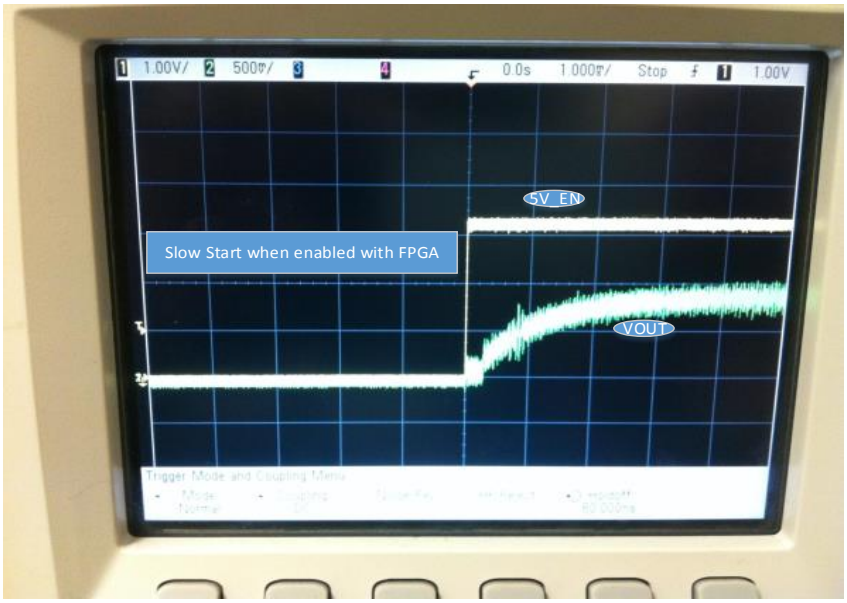
Shown below is the Efficiency vs Load in PWM mode. The Efficiency drops below 500mA load approximately. Also, it can be seen that R_{DCR} value of 50mΩ drags down the efficiency by 10%-15% approximately. This is expected because the output power of this Buck Converter is low since the output voltage is only 0.9V.



42. Efficiency vs. Load in PWM Mode

6.2.6 Start-Up

The Start-up is done be enabling the chip from FPGA since at the same time Digital Controller needs to be reset. The use of FPGA can be avoided if solution described in Future Work section is implemented on-chip in the next revision. Shown in the next page is the result.



43. Slow Start in PWM Mode

7 CONCLUSION & FUTURE WORK

7.1 Conclusion

A digital controller is designed using s - domain method to regulate the output voltage. A PFM detector is designed to detect PFM mode and boost converter's efficiency. Besides, SPI and high speed serial interface are designed to program, characterize mixed signal blocks and debug the chip. Also, control and clock gating logic is implemented to save power by turning off the blocks when not in use. All of the above blocks are implemented in IBM 0.18 um CMOS process. The area of the digital core is 0.6mm^2. A test board is built with the chip, power FETs and driver IC. The PWM mode is validated till 9A load currents (at the time of this writing). Also, different controller coefficients are computed as per the UGB, ESR etc., and programmed to optimize the dynamic response. All mixed signal blocks are characterized through high speed serial interface. The functionalities of PFM detector, slow start, mode transitions between PWM and PFM, SPI and high speed serial interface are also verified on the board.

7.2 Future Work

7.2.1 Size of the Digital Controller

During the evaluation of PFM to PWM transition on the board, a high ESR is needed to let the detector circuit kick in the transition. However, this high ESR is affecting the steady state settling in PWM mode despite new coefficients are programmed for the corresponding ESR. This issue is replicated in Matlab as well. After investigating the root

cause of the issue – which most likely is going to be the effect of rounding noise in the digital controller – the size of the coefficients may have to be increased or architecture of digital controller may have to be tweaked.

7.2.2 Reset Digital Controller on Start-Up

When the chip is powered up and slow start is disabled, ideally, the error signal going into digital controller should be zero since inputs to reference and output ADCs are zero. However, it is noticed during chip evaluation that the digital controller is saturated before enabling slow start due to non-zero error signal. This problem is solved by resetting digital controller using a FPGA board just before enabling slow start. In the next revision this functionality can be moved to on-chip.

7.2.3 Controller Discretization Methods

For designing the Controller in this thesis, Bi-linear transformation is used to discretize the continuous time domain controller. Though the transient simulation results achieved with this digital controller are comparable with that of its analog domain counterpart, this digital controller seemed to not support smooth transition between PFM and PWM modes in simulations. So, it is worth investigating other discretization methods followed by modelling and analysis in Matlab, and finally implementing in the chip's next revision.

REFERENCES

[1] R. Lenk, "Application bulletin AB-20 optimum current-sensing techniques in CPU Converters," *Fairchild Semiconductor Application Notes*, 1999.

[2] P. Midya, M Greuel and P. Krein, "Sensorless Current Mode Control - An Observer Technique for DC- DC Converters," *IEEE Trans. Power Electronics*, vol. 16, pp. 522–526, July 2001.

[3] H. P. Forghani-zadeh and G. A. Rincón-Mora, "Current-sensing techniques for DC-DC converters," *IEEE Midwest Symp. on Circuits and Systems*, pp. 577–580. Aug. 2002.

[4] T. Liu, H. Yeom, B. Vermeire, P. Adell, and B. Bakkaloglu, "A Digitally Controlled DC-DC Buck Converter with Lossless Load-Current Sensing and BIST Functionality," in *IEEE ISSCC Dig.,* 2011, pp. 388–398.

[5] L. Guo, J. Y. Hung, and R. M. Nelms, "Digital controller design for buck and boost converters using root locus techniques," in *Proc. IEEE Ind. Electron. Conf.,* Nov. 2003, pp 1864-1869.

[6] L. Guo, "Implementation of Digital PID Controllers for DC-DC Converters using Digital Signal Processors," in *Proc. IEEE Electro/Information Technology Conf. (EIT),* pp. 306–311, May 2007

[7] A. Prodic, D. Maksimovic, and R. Erickson, "Design and implementation of a digital PWM controller for a high-frequency switching dc-dc power converter," in *Proc. 27th Annu. Conf. IEEE Ind. Electron. Soc.,* 2001, vol. 2, pp. 893–898.

[8] A. V. Peterchev and S. R. Sanders, "Quantization resolution and limit cycling in digitally controlled PWM converters", *IEEE Trans. Power Electron.,* vol. 18, no. 1, pp. 301–308, Jan. 2003.

[9] R. Erickson, and D. Maksimovic, "Fundamentals of Power Electronics".

[10] X. Zhang and D. Maksimovic, "Digital PWM/PFM controller with input voltage feed-forward for synchronous buck converters," in *Proc. IEEE Applied Power Electronics Conference and Exposition (APEC),* Feb. 2008, pp. 523–528.

[11] P. Midya, M Greuel and P. Krein, "Sensorless Current Mode Control-An Observer Technique for DC- DC Conveners", *IEEE Trans. Power Electronics*, vol. 16, pp. 522 -526, July 2001.

[12] "Design Compiler User Guide", Synopsys Inc.

[13] “Synopsys Timing Constraints and Optimization User Guide”, Synopsys Inc.

[14] “Encounter User Guide”, Cadence Design Systems.