uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Zhiyuan Hu

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Systems Science)

GRADE / DEGREE

Department of System Science

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A Multiplexed Feedback Solution for Isolated Digital Power Supplies

TITRE DE LA THÈSE / TITLE OF THESIS

Tet Yeap

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Yan-Fei Liu

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

Voicu Groza

Ahmed Karmouch

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

A Multiplexed Feedback Solution for Isolated Digital Power Supplies

Zhiyuan Hu

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the M.Sc. degree in Systems Science

Systems Science
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-65552-8
Our file Notre référence
ISBN: 978-0-494-65552-8

Table of Contents

List of Figures

Legend

Acronym	Meaning
ADC	Analog-to-digital converter
CAN	Controller Area Network
DPWM	Digital pulse-width modulator
I^2C	Inter-Integrated Circuit
LIN	Local Interconnect Network
LSB	Least significant bit
LUT	Lookup table
LVDS	Low Voltage Differential Signaling
OEM	Original Equipment Manufacturer
PID control	Proportional–integral–derivative control
PLL	Phase-lock loop
PMBus	Power Management Bus
PWL	Piece-wise linear
PWM	Pulse-width modulator
RD	Running Disparity
RZ	Bipolar return-to-zero
SMBus	System Management Bus
SST	Simple Serial Transport
UART	Universal Asynchronous Receiver/Transmitter

Abstract

As digital power supply became a significant market opportunity, manufacturers started looking at ways to control the system cost by reducing components and their technical requirements.

This thesis proposes a multiplexed digital feedback solution for isolated digital power supply systems. Several digital isolators used in existing applications can be reduced to one. The data transmission speed is greatly reduced. The communication protocol is DC-balanced; therefore pulse-transformer type digital isolators can be used resulting in further lower cost. The proposed solution also improves output accuracy over existing solutions.

Presentation of the proposed system will be in three parts: architecture, protocol, and Piece-wise Linear Feedback method.

Both theoretical analysis and practical implementation will be discussed, and the simulation results included support the theoretical analysis. A prototype board was built and its performance is promising.

Acknowledgements

I am heartily thankful to my supervisors, Dr. Tet Yeap, Dr. Yan-Fei Liu, and Mr. Raymond Orr. Without their supervision and warmhearted support, this thesis would not have been possible.

Dr. Tet Yeap agreeably accepted me as his student so that I could have the opportunity to write this thesis in partial fulfillment of the requirements for my Master's Degree. Dr. Yeap provided me important support throughout the research process and systematic guidance for organizing this thesis.

I owe my deepest gratitude to Dr. Yan-Fei Liu for his crucial contribution to my education in power electronics and his great involvement in this thesis research. He provided academic opinions at every step of this research, and to this thesis. His strict requirements and friendly encouragement guided me onto a path towards excellence.

I wish to express my sincere appreciation to Mr. Raymond Orr for his guidance and help in the past years. He offered me the work opportunity in the industry and led me into the world of power electronics. The research idea was enlightened by his insight of market; and the many discussions we had during the research progress were fundamental to this thesis.

I deeply appreciate my managers and colleagues in Power Integrations Canada ULC for their helps for my studies. Without the skills and experience that I obtained from them, the implementation part of this project would not have been successful. The spirit of precision and dedication that I learnt from them will benefit me in my future study and career.

My colleague John Estabrooks has helped me with the magnetic design and offered me his precious engineering experience. My colleague Tony Reinberger has helped me in numerous ways, from sharing his talented knowledge and skills in electronics to offering me rides in stormy nights. Ted Carron has been a dependable source of electronic components when I was in dire need of them during circuit debugging. David LeBlanc kindly offered me his racing bike in support of my enjoyable commute. My manager Roger Colbeck helped me correct some mistakes in this thesis.

Power Integrations Canada ULC has funded my studies and provided the equipment.

I also would like to thank Roger Miskowicz for verbal corrections and his pertinent suggestions.

Many thanks also go to Ms. Monique Walker for her perfect administration work. Her kindhearted help was very important to me on the road towards success.

I am deeply thankful to my parents for their huge supports.

This thesis is a tribute to my beloved grandmother.

Zhiyuan Hu

Chapter 1 Introduction

As the embedded computing technology becomes mature and the cost drops in accordance with Moore's law, digitally controlled power supplies are becoming a significant market opportunity [1][2]. According to a questionnaire developed by Linnea Brush [2], "nearly 60% of respondents expect to use digital power management in the future," where the "respondents" consist of "semiconductor manufacturers, power converter makers and system OEMs."

Digital power supplies promise improved performance by applying advanced digital control techniques such as digital proportional-integral-derivative (PID) control [3][4][5], fuzzy logic control [6][7], and other novel non-linear control strategies [8] into power supply systems. It reduces the amount of external components, which reduces the system cost, and in-system programmability, which increases the system flexibility [9] [10]. Because of these benefits, many investigations have been done by both academia and industry in order to bring digital power into the market.

This thesis investigates the data communication in isolated digital power supplies, which has not been systematically addressed before. The thesis proposed and implemented a multiplexed feedback solution, optimized for isolated digital power supply systems. This

solution consists of a data communication bus, a feature-rich yet efficient protocol, and a data compression method. A prototype power supply system with two Flyback Converters was built to validate the effectiveness of the proposed system.

1.1 Motivations

Power supplies can be divided into two broad categories: isolated and non-isolated power supplies. For safety reasons, many electric appliances that draw power from an AC line isolate the output side (secondary side) from the input side (primary side). This isolation prevents any electric connection between the two sides, and therefore, preventing signal transmission. Traditionally in analog power supplies, the feedback signals are transmitted by linear opto-couplers, as shown in Figure 1-1[11]. Linear opto-couplers provide isolation while transmitting analog signals from secondary side to primary side. In digital power supplies, however, signals are digital and therefore linear opto-couplers are not fast enough for digital communications.

Logic opto-couplers and pulse-transformers are commonly used high-speed isolators, whose bandwidths are normally tens of mega bits per second. A few opto-couplers (e.g. Agilent HCPL901J) and RF-coupler isolators (e.g. Silicon Laboratories SI8442-C-IS) can

achieve over 100Mbps but they are fairly expensive. The latest research conducted by Baoxing Chen extended this record to 200 MHz[12][13][14].

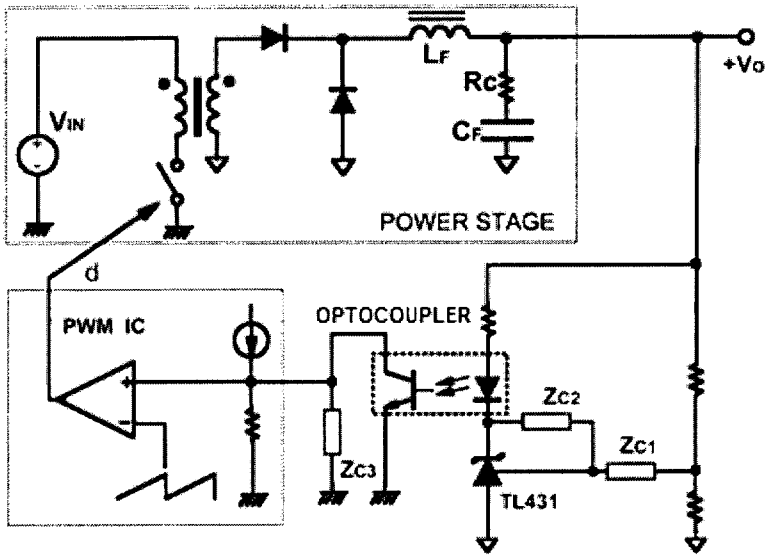


Figure 1-1 Simplified circuit diagram of isolated power supply with optocoupler feedback, from[11]

Cutting-edge Switching Mode Power Supplies (SMPS) switch at hundreds of or thousands of kilo hertz, where a large volume of sample data are created and therefore costly high-speed isolators must be used. What's more, many electric applications contain multiple isolated power supply modules, such as Plasma TVs, LCD TVs, medical equipment, bench top power supplies, and so on, and each power module uses several high-speed isolators to transmit digital samples and control functions such as On/Off and Over Voltage/Load/Temperature Protection. This will quickly add up the cost to more than make sense.

The motivation of this thesis is reducing the system cost on high-speed digital isolators, although as it will show in later chapters, the performance can be improved at the same time.

1.2 Thesis Objectives

The design of the proposed system will follow five approaches below to reduce the system cost on high-speed digital isolators.

Approach 1

Multiplexing the digital feedback loops into a single communication bus will reduce the number of digital isolators used in the system thus reduce the system cost.

Approach 2

Multiplexing control functions that will otherwise use separate isolators, such as "On/Off", "Over Current Protection", "Over Temperature Protection", etc., into the data communication bus will reduce the total number of digital isolators.

Approach 3

An efficient communication protocol will minimize the framing bits, thus lower the data transmission speed allowing cheap digital isolators to be used.

A properly designed protocol will also reduce the number of signal lines, such as the clock line and the frame synchronization line, allowing less digital isolators to be used.

Approach 4

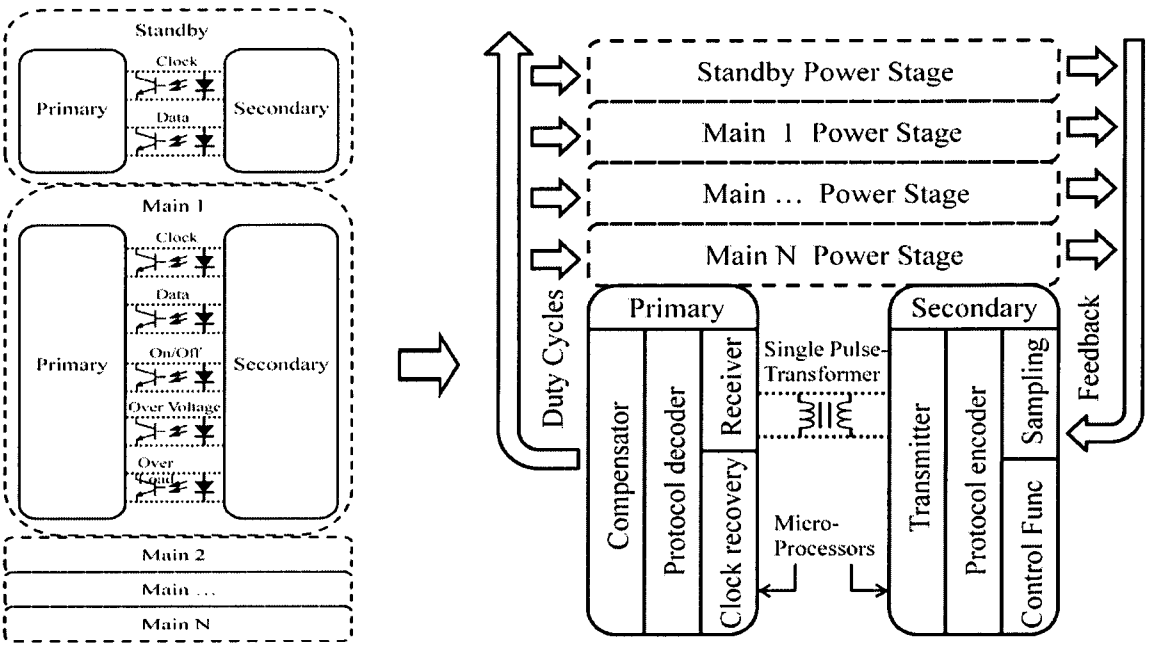
An efficient encoding strategy will reduce the number of bits in each sample, thus lower the speed requirements, allow cheap digital isolators to be used, and thus reduce the system cost.

If a 10-bit ADC is used in a digital power supply, each sample consists of 10 bits, plus the extra framing bits. Encoding each sample into fewer bits will reduce the data bit rate resulting in simpler design, lower power consumption of the digital circuitry, and reduced system cost.

Approach 5

Using pulse-transformers, whose volume price is tens of cents, instead of high-speed opto-couplers or RF-couplers, whose prices are around $2 to $10 dollars, will significantly reduce the system cost. However, the use of pulse-transformers requires the data stream to be DC-balanced in order to prevent inductance saturation.

As a result of above discussions, this thesis aims at designing a multiplexed feedback solution for isolated digital power supplies, as shown in Figure 1-2, which greatly saves the system cost on digital isolators. This solution will include a multiplexed system architecture, an efficient and feature-rich protocol, and a data compression method.



(a) Traditional System

(b) Object System

Figure 1-2 The object system saves cost on digital isolators

The overall benefits, on top of the reduced cost on digital isolators, also include improved output accuracy and lowered data transmission speed which results in lowered power consumption of the digital circuitry. The multiplexed communication bus also reserves space for future compensation methods.

This thesis will discuss the communication issues and the proposed protocol in detail, analyze the compression method and its effect on dynamic response theoretically and validated by simulation, and build a prototype to prove the overall system.

1.3 Thesis Contributions

This thesis systematically addressed and optimized the communication issues in isolated digital power supply systems in order to provide a solution to reduce the system cost.

This thesis proposed and implemented a multiplexed feedback architecture for isolated digital power supplies, which reduces the number of digital isolators.

This thesis designed a protocol that is optimized for the multiplexed real-time communication in isolated digital power supplies, and is significantly more efficient than other existing protocols used in isolated digital power supplies. It's the first protocol that uses 8B/10B line-code in power supplies for real-time communications which allows the elimination of clock line and frame synchronization as well as some address bits. Some control characters are also shorter than those implemented in higher level protocols. This reduced the speed requirement for digital isolators thus reduced the system cost.

The Piece-wise Linear Feedback method is used in power supplies for the first time for data compression. It uses only 4 bits to present a sample while retaining the accuracy of a 10-bit ADC without compromising the dynamic response. This reduced the speed requirement for digital isolators as well, and thus reduced the system cost.

The engineering implementation in this thesis involved considerable amount of work. This work provides a platform for future studies and improvements.

1.4 Chapter Organizations

Chapter 2 introduces the background knowledge in power supplies and the related work done by others. Chapter 3 describes the proposed system and discusses the considerations in protocol design. Chapter 4 discusses the Piece-wise Linear Feedback method used for data compression, and its impact on dynamic response. A design example is given and is verified by simulation. Chapter 5 describes the engineering implementation. Chapter 6 presents the experimental performance. Chapter 7 draws conclusions and suggests future work.

Chapter 2 Background

2.1 Introduction

Building a power supply system requires knowledge in both power circuitry and control theory. Building a digitally controlled power supply requires additional considerations and knowledge. This chapter summarizes the background knowledge that will be used in the proposed system. Section 2.2 reviews the Flyback topology. Section 2.3 reviews the feedback control in Flyback topology. Section 2.4 reviews the control techniques in digital power supplies. Section 2.5 reviews the implementation of Digital PWMs and related issues. A literature review of previous related work is provided in Section 2.6.

2.2 Flyback Converter Topology

Switching Mode Power Supplies have many topologies, such as Buck, Boost, Push-Pull, Half/Full-Bridge, and so on. They have different principles using magnetic and electronic components to transform energy from the input voltage to the desired output voltage.

Flyback topology is widely used for delivering power from 5W to 150W [15]. A typical Flyback converter topology is shown in Figure 2-1[16].

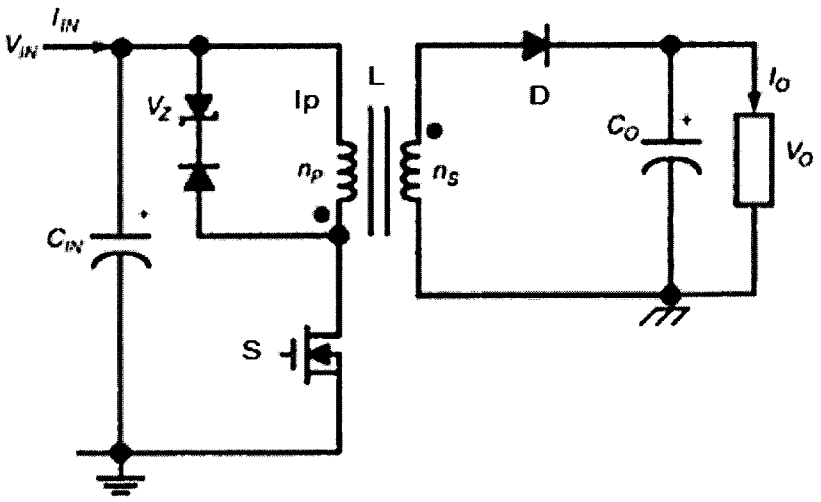


Figure 2-1 Schematic of a Flyback converter, from [16]

In Figure 2-1, V_{in} is the input DC voltage. L is the primary inductance. S is the power MOSFET, which is controlled by a pulse-width modulator (PWM) (not shown). I_p is the primary current. D is a rectifier diode. C_o is the output capacitor. R is a resistive load. V_o is the output voltage.

When the MOSFET is turned on, the transformer's primary winding is connected between V_{in} and ground. The rectifier diode, D, is reverse-biased allowing no current to flow through the secondary winding. The primary winding behaves like an inductor and stores energy. The primary current can be calculated as $I_p = (V_{in} - 1)T_{on} / L$, where 1 represents the approximate voltage drop on the MOSFET. The energy stored in primary

winding is $E = \frac{LI_p^2}{2}$. During this time, no energy is delivered from the primary side. The output capacitor C_o supplies the current to the load.

When the MOSFET is turned off, the polarity of the primary winding reverses, causing the rectifier diode to be forward biased. The secondary winding dumps energy E to the secondary-side circuit, simultaneously charging the output capacitor C_o and supplying the load current.

The power MOSFET is controlled by a PWM. A feedback signal from the output voltage V_o is used to calculate the pulse width T_{on}. When V_o is higher than the reference voltage, T_{on} decreases, so that less energy can be delivered to the secondary side. When V_o is lower than the reference voltage, T_{on} increases, and energy is delivered to the secondary side.

2.3 Feedback-Loop Stabilization

Switching Mode Power Supply is an atypical feedback system: the power stage is controlled discretely using PWM, while its output is averaged by a filter for continuous power delivery. The flow graph of power supply control system is shown in Figure 2-2[17].

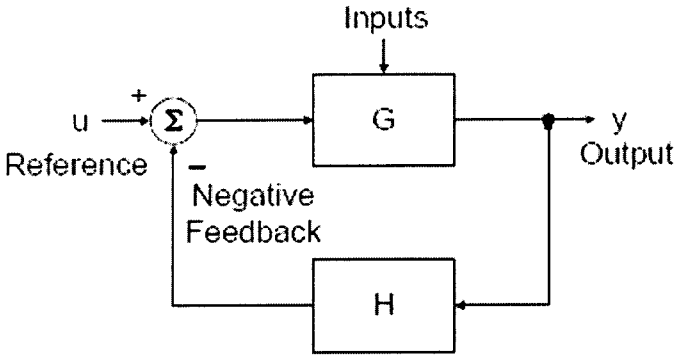


Figure 2-2 Flow Graph of Power Supply Control, from [17]

In Figure 2-2, G is the power circuitry and H is the control circuitry. The inputs for the system include power and reference sources. The operating point of the system is set by the reference source, u.

The control process can be described as follows: The output voltage is divided down by a resistive divider, compared with the reference voltage (the "u" in Figure 2-2 and the "DC Reference" in Figure 2-3[17]), and the error signal is amplified and compensated by an error amplifier network. The compensated signal is sent into a PWM. In the PWM, the compensated signal is compared with a sawtooth waveform and generates square waves to drive the power transistor. The width of the pulses is from the beginning of each sawtooth cycle to the point where the sawtooth voltage equals to the compensated the error signal voltage. The next cycle begins with the waveform restored to the beginning of the sawtooth.

When a load draws power from the power supply, the output voltage drops and an error voltage V_E is generated. This increases the pulse width to deliver more energy from the source to the load.

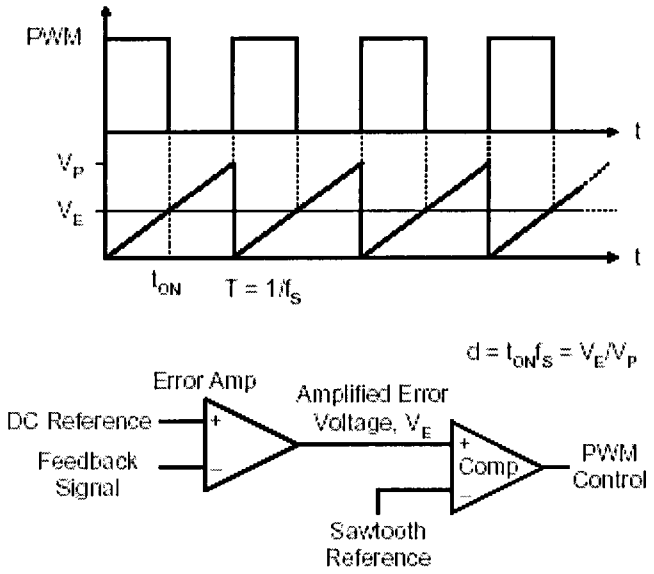


Figure 2-3 Typical PWM control implementation, from [32]

Since the PWM is discrete its frequency must be sufficiently higher than the loop bandwidth so that the system can be considered continuous at the frequency of interest.

Once the small-signal model of the power supply system has been derived the Bode Plot technique can be used for stability analysis.

The general stability criteria can be represented by gain and phase margins. The recommended gain margin value is from -6dB to -12dB, and the recommended phase margin value is between 45° to 60°. To meet these criteria the gain must be -20 dB/decade

at the cross-over point otherwise a compensator will be required to stabilize the closed-loop system.

One commonly used compensation technique for Flyback converters is Phase-Lag compensation. Its implementation in an analog circuit is shown in Figure 2-4 [17]. It adds a pole at a low frequency in order to get a good phase margin at the cross-over point. The Bode plot is shown in Figure 2-5 [17].

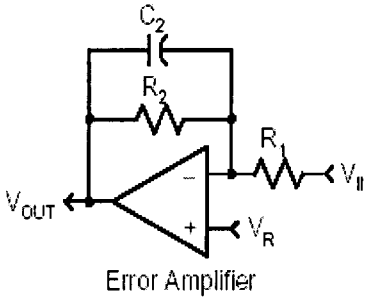


Figure 2-4 Phase-Lag compensation, from [17].

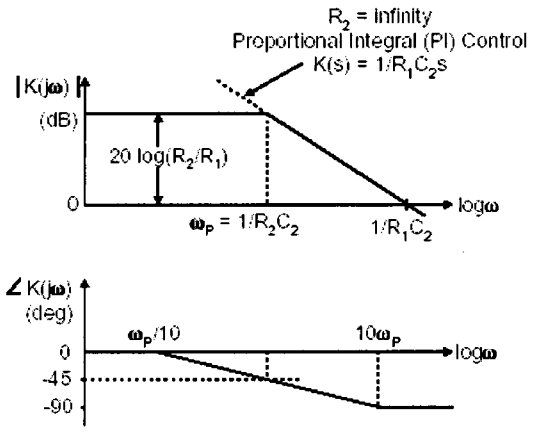


Figure 2-5 Bode plot of Phase Lag compensation, from [17]

2.4 Digital Control in Power Supplies

PID control is widely used for system stabilization as discovered from literature searches for digital switching-mode power supply controllers.

The transfer function of a power supply system should be continuous. While designing a digital compensator, two methods were investigated: an Analog redesign and Direct-

digital design. Using analog redesign method, the first step was to develop a PID controller in continuous domain, and then transform it into discrete domain. The transformation is based on mathematical approximations, therefore, errors are inevitable. Direct-digital design method studied the effects of digital poles and zeros on Bode plot so that the results can be directly applied in continuous domain.

The most recent contributions have been made by H. Al-Atrash and Shamim Choudhury in 2007. Their works are briefly described in this section.

Direct-digital Design

H. Al-Atrash studied digital zeros in analog domain [4]. He mapped the gain and phase characteristics of digital zeros into Bode plot. His work is briefly described below.

(1) Single (first order) zeros

Single zeros can be represented as:

$$H_{zero}(z) = 1 - (1 - 1/a)z^{-1} \tag{2.1}$$

where a is a constant.

Single zeros can boost the phase up to 90 degrees, and increase the gain to 20dB/decade. The Bode plot is shown in Figure 2-6.

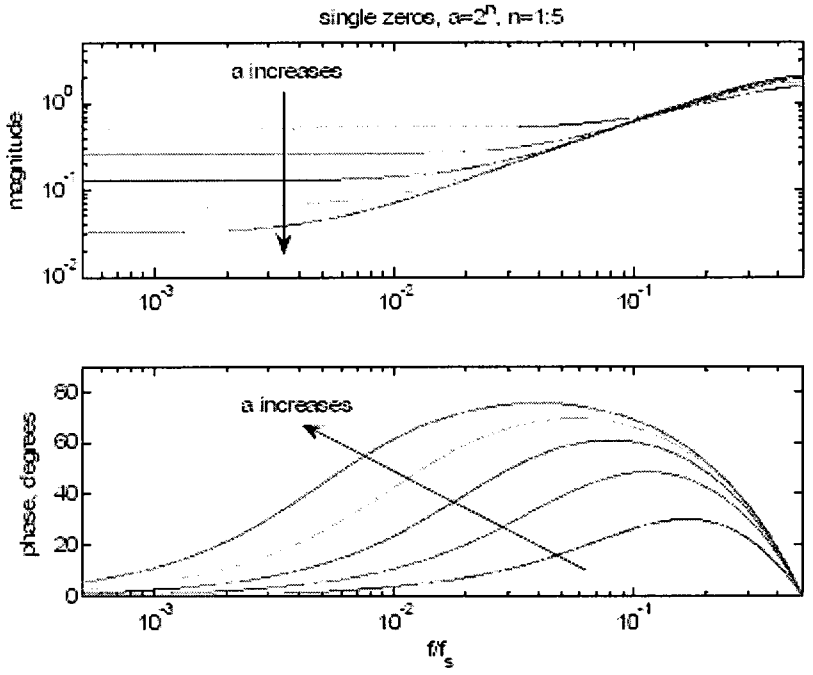


Figure 2-6: Single zeros, from [4]

(2) "Hard" complex zero-pairs

"Hard" complex zero-pairs can be represented as:

$$H_{ZeroPair}^{Hard}(z) = 1-(2-1/b)z^{-1}+z^{-2} \tag{2.2}$$

where b is a constant.

"Hard" complex zero-pairs can boost phase with a step of 180 degrees, and increase gain approaching 40dB/decade. The Bode plot is shown in Figure 2-7.

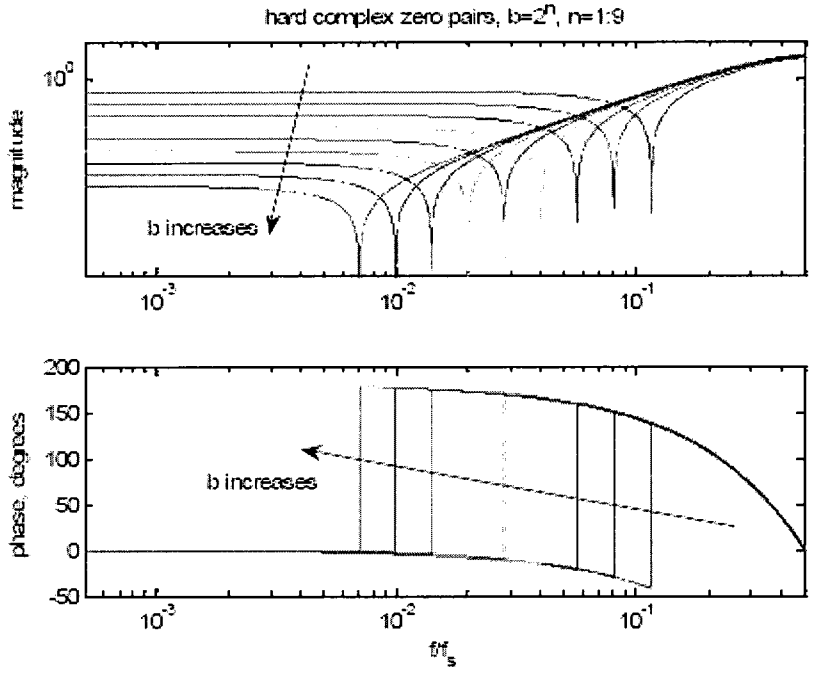


Figure 2-7: "Hard" complex zero-pairs, from [4]

(3) "Soft" complex zero-pairs

"Soft" complex zero-pairs can be represented as:

$$H_{ZeroPair}^{Soft}(z) = 1-(2-1/b)z^{-1}+(1-1/c)z^{-2} \tag{2.3}$$

where b and c are constants.

"Soft" complex zero-pairs can boost phase up to 180 degrees, and increase gain to 40dB/decade. The Bode plot is shown in Figure 2-8.

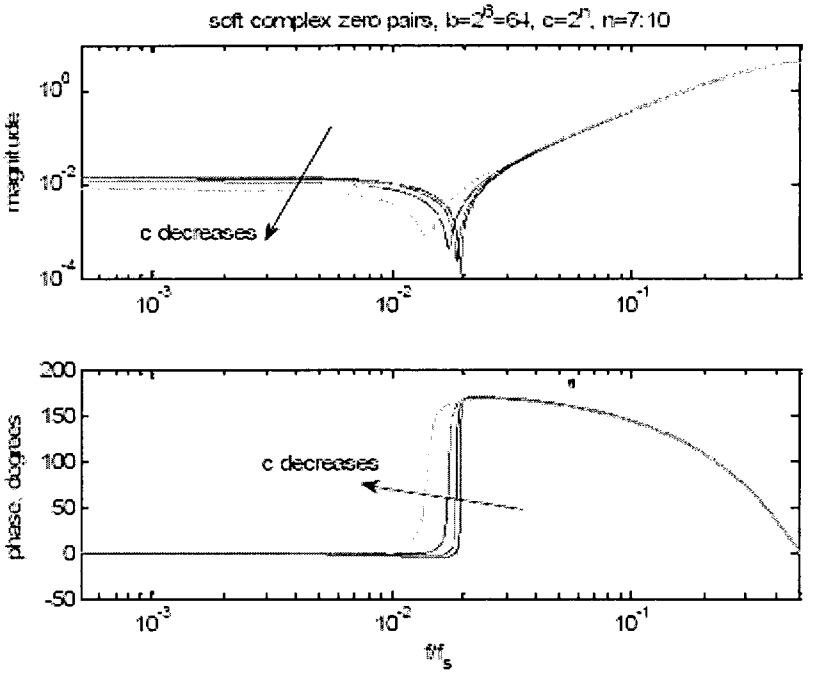


Figure 2-8 "Soft" complex zero-pairs, from [4]

(4) Euler integrator

Euler integrator can be represented as:

$$H_{Integrator}^{Euler}(z) = \frac{1}{1 - z^{-1}} \tag{2.4}$$

Euler integrator increases gain in low-frequency so that it can minimize steady-state error. It delays phase by 90 degrees. A limiter function is required to avoid the integrator overflow. The Bode plot is shown in Figure 2-9

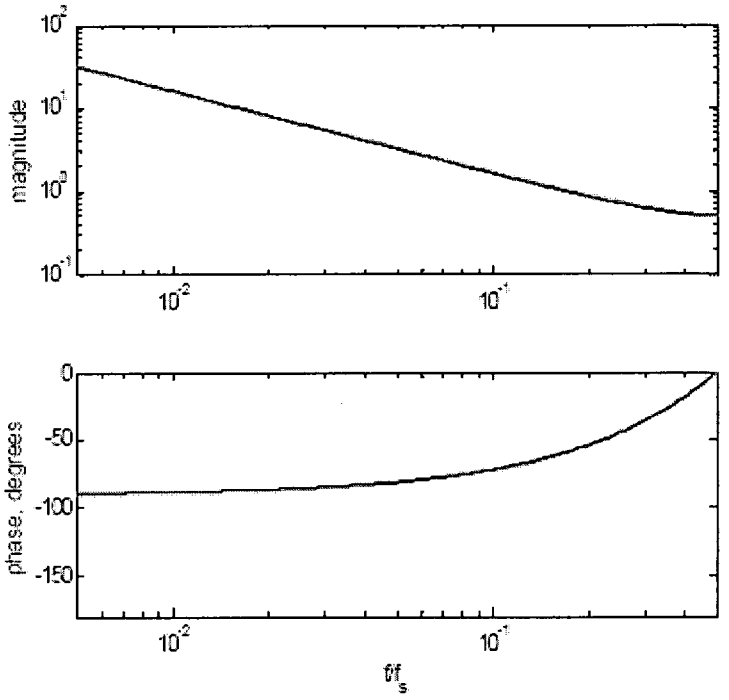


Figure 2-9 Euler integrator, from [4]

Based on H. Al-Atrash's work, direct-digital designs can be transformed directly into analog domain, using familiar tools without obtaining a discrete-time model of the power stage.

Analog Redesign

Shamim Choudhury's work is based on analog redesign [5]. He provided a detailed design procedure for digital power supplies and examples. The thought behind the design procedure is described below:

Once s-domain compensator design has been performed the compensator $G_c(s)$ is in the form of:

$$G_c(s) = \frac{A}{s}(s - s_1)(s - s_2) \tag{2.5}$$

where s_1 and s_2 are s-domain zeros.

Then consider the desired z-domain compensator function $G_c(z)$, which must be in the form of:

$$G_c(z) = K_c \frac{(z - z_1)(z - z_2)}{z(z - 1)} \tag{2.6}$$

where z_1 and z_2 are z-domain zeros.

Because $z_1 = e^{s_1 T_s}, z_2 = e^{s_2 T_s}$ where T_s is the sampling period, K_c is the only unknown factor in $G_c(z)$.

Noting that $G_c(z) = G_c(s)$ are at the desired loop crossover frequency f_{cov}, where $s = s_{\text{cov}} = j2\pi f_{\text{cov}}, z = z_{\text{cov}} = e^{j2\pi f_{\text{cov}} T_s}$, we can finally get K_c.

This method simplified the problem by avoiding the using of z-transforms, while guaranteeing the desired crossover frequency will not shift, which might otherwise be caused by z-transform approximations.

2.5 Digital PWM

Digital Oscillator

An implementation of a simplified digital oscillator is shown in Figure 2-10 [10].

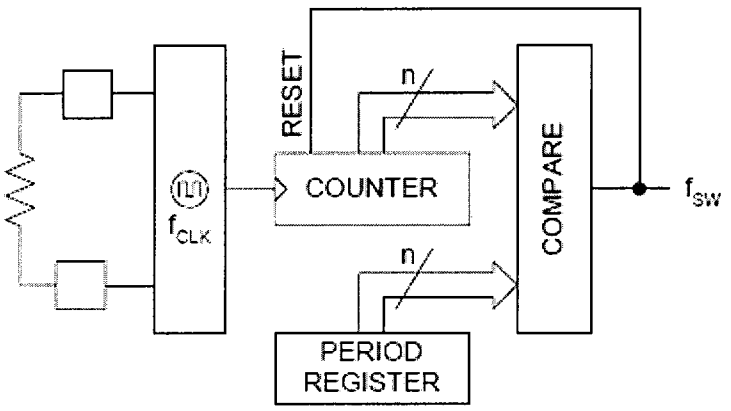


Figure 2-10: Simplified digital oscillator, from [10].

An up counter is driven by an internal clock. The counts are synchronously compared with the period register. Every time the two values are equal, the output signal f_{sw} flips and resets the counter. f_{sw} is the PWM frequency that is preset by the Period Register.

Duty-cycle resolution

Duty-cycle resolution is the percentage of each internal timer count period in a PWM switching cycle, as shown in Figure 2-11 [3].

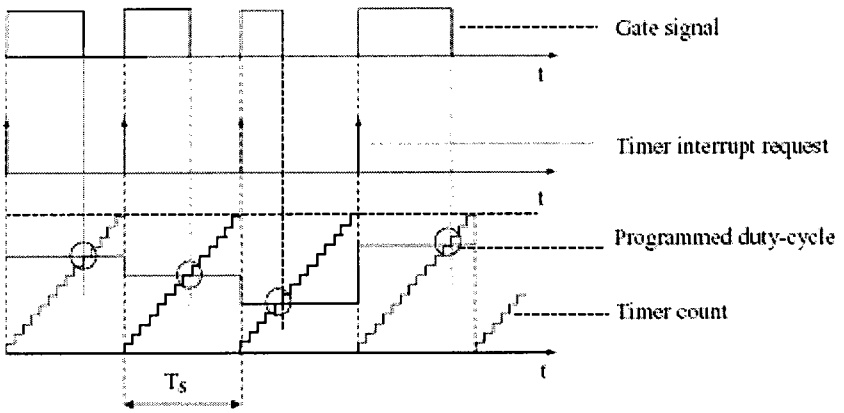


Figure 2-11: Duty-cycle resolution, from [3]

Duty-cycle resolution is the key factor of the accuracy of the digital pulse width modulation (DPWM). If the Duty-cycle resolution is coarse, the power supply may be subjected to Limit Cycle Oscillation which will be discussed in Section 2.5.

Duty-cycle resolution of DPWM is defined as

$$D_{RES}(\%) = \frac{f_{sw}}{f_{clk}} \times 100\% \tag{2.7}$$

An analog PWM has effectively unlimited resolution.

ADC resolution

ADC is used in digital controlled power supply to sample the output voltage. The resolution of an ADC can be calculated to be approximately [3]:

$$res_{ADC} = \frac{V_{ref}}{2^n} \tag{2.8}$$

where n is the number of bits and Vref is the reference voltage of the ADC.

ADC resolution is the key factor in output voltage stability. If the ADC resolution is very fine compared to the Duty-cycle resolution, the power supply may be subjected to Limit Cycle Oscillation that will be discussed in Section 2.5.

Limit Cycle Oscillation

Limit Cycle Oscillation is caused by a too coarse Duty-cycle resolution with respect to the ADC resolution, regardless of whether the design meets all conventional stability criteria, as shown in Figure 2-12[18].

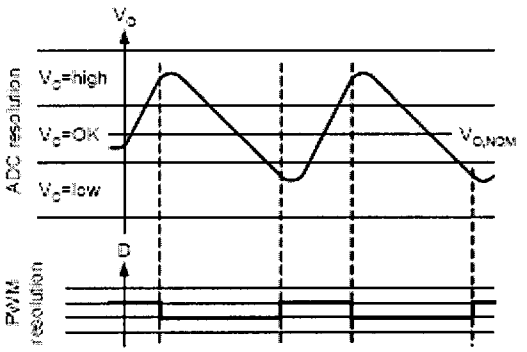


Figure 2-12: Limited Cycle Oscillation, from [18]

Figure 2-12 shows the mechanism of Limited Cycle Oscillation. It can be seen that ADC has a finer resolution than PWM. When the ADC detects the output voltage is one

LSB higher than the "V_o=OK" range, it reduces PWM pulse width by 1 LSB. However, because the PWM resolution is too coarse, the output voltage drops into the "V_o=low" range. The PWM then increases the pulse width by 1 LSB, and then the output voltage rises back to "V_o=high" range. This circle will repeat again and again, and the output voltage starts to oscillate.

According to Peterchev[18], the following three conditions must be met in order to avoid Limited Cycle Oscillation:

(1) resolution (DPWM) > resolution (ADC); typically N(dpwm)=N(adc)+1, where N is the number of bits;

(2) 0< Ki < 1, where Ki is the integral term of PI controller;

(3) $1+N(A)L(j\omega)\neq 0$ (Nyquist Criterion), where A is the AC amplitude of output voltage, L is the transfer function from ADC to output voltage.

2.6 Previous Work on Data Communications in Isolated Digital Power Supply

Many protocols (in both physical layer and data link layer) used in digital power management include System Management Bus (SMBus)[19][20], Power Management Bus (PMBus)[21], Simple Serial Transport (SST)[22][23], Universal Asynchronous

Receiver/Transmitter (UART)[24], Inter-Integrated Circuit (I^2C)[25], Controller Area Network (CAN) [26][27], Local Interconnect Network (LIN)[28], and so on. However, most of them are designed for "event management" in large power supply systems, e.g., passing a "Failure" event from chassis to chassis, shelf to shelf, or even across campus [29]. These protocols are mostly in low speed, or add many framing bits into the data stream. Some studies suggest these protocols are not suitable for transmitting real-time data for feedback control [30][29][31].

Meanwhile, the isolated digital power supplies that have been developed so far use inefficient protocols [32] [33]. Some of them transmit pulse width modulated (PWM) duty cycle rather than data [9] [33]. These designs compromise output accuracy or dynamic response, and failed to optimize the system as a whole. Nevertheless, it is helpful to review those methods in detail before introducing the optimized solution.

Existing communication methods in isolated digital power supplies can be summarized into two types: transmitting duty cycle information and transmitting feedback data.

Ka Leung's Method

Ka Leung and Don Alfano designed a 400 KHz isolated digital power supply in 2006[9], using a secondary controller and transmitting duty cycles to the primary side via a 150 MHz quad-channel digital isolator[34]. The primary microcontroller also transmits

feed-forward data to the secondary side controller via other channels in the same digital isolator.

This design (shown in Figure 2-13) takes full advantages of the quad-channel digital isolator, yet it has a number of limitations.

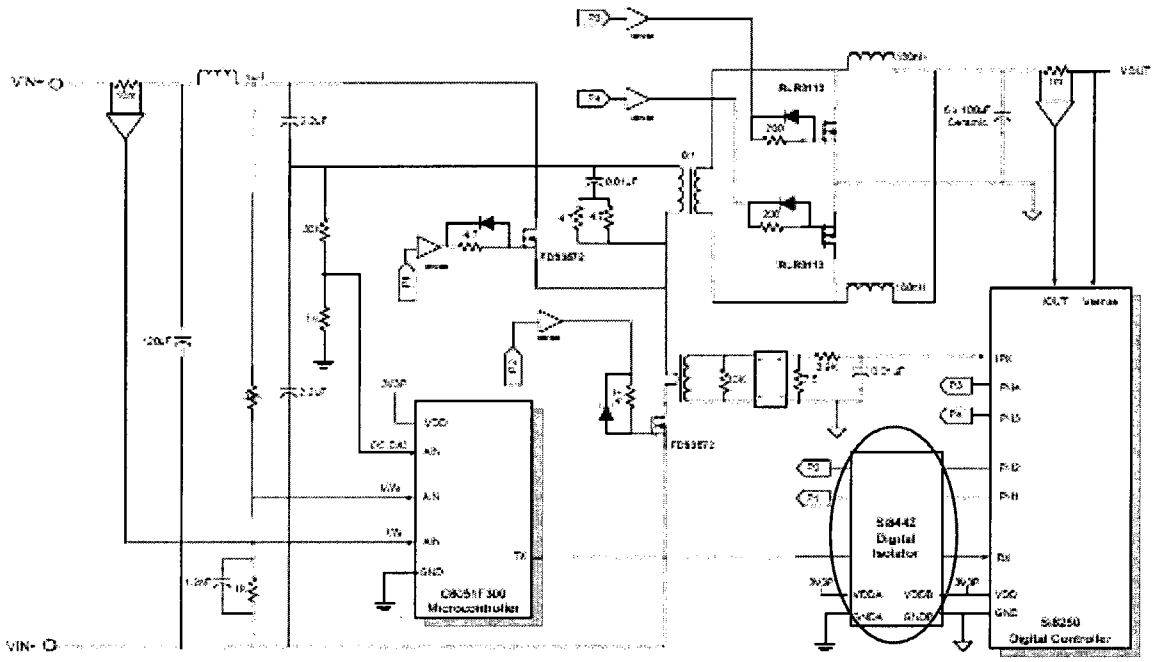


Figure 2-13 Diagram of Ka Leung's system, from [9]

First of all, although the digital isolator is almost the fastest one in the market, it is still not fast enough to provide satisfactory duty-cycle resolution. For a 400 KHz switching frequency design, the duty-cycle resolution through the digital isolator is only $\frac{400\times10^3}{150\times10^6}\times100\%=0.267\%$; while a DPWM with a 150 MHz internal clock and a 100-tap delay line requires a resolution 100 times finer than that. What's more, if the system uses a

10-bit ADC, the duty-cycle resolution is 2.734 times coarser than the ADC resolution, which may result in limit-cycle oscillation. In fact, the ADC used in this system is only 6-bit. This system's output accuracy is significantly lower than other designs.

Second, the duty cycles cannot be multiplexed therefore requires additional digital isolators and thus increases the cost.

Third, the protocol used in this design is not described in this paper. It is not clear if the protocol supports multiplexed use of the data bus. Nevertheless, for a 150 MHz isolator and a 400 KHz PWM, 375 bits per cycle is far more than needed.

Aleksandar Prodic's Method

Aleksandar Prodic et al. used a different approach for isolated communication [32]. His architecture is shown in Figure 2-14.

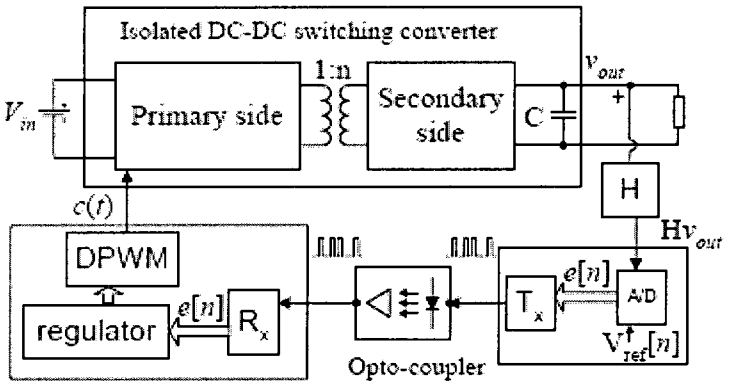


Figure 2-14 Diagram of Aleksandar's System, from [32]

The controller was designed for a single-stage power supply. In this system, two high-speed opto-couplers are used for separate data and the clock lines.

As shown in Figure 2-15, only 4 LSB of an 8-bit error are transmitted to the primary side. Any error value above or below this range are replaced by 1111 and 0000, respectively. This method reduces the requirement of the communication bandwidth; however its accuracy is compromised: in this system, 1 LSB of the ADC represents 0.5% of the output voltage and therefore 4 LSB covers ±4% of the output voltage range. In many applications, 0.2% output accuracy is required; in that case, 4 LSB can only cover ±1.6% output voltage range and that is certainly not enough for dynamic response.

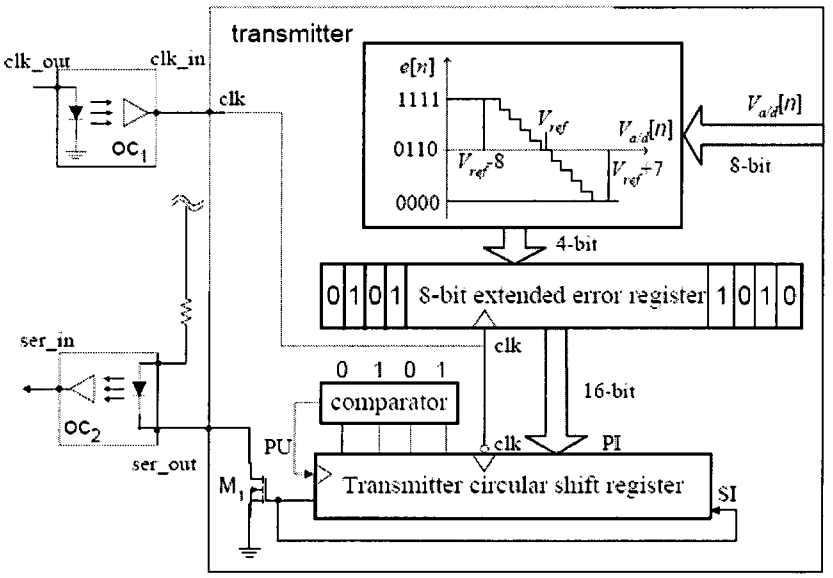


Figure 2-15 Diagram of the serial transmitter, from [32]

The frame synchronization strategy used in this system is to extend the 4-bit error value to 8 bits by doubling each bit (for example, extend 0110 to 00111100), and then add a start sequence (0101) at the beginning of each frame and a stop sequence (1010) at the end of each frame. Every data bit is doubled, and only framing bits are singular, thus data bits can be distinguished from framing bits. Therefore in this method, the effective amount of bits transmitted in a 16-bit frame is only 4. Therefore the efficiency of this protocol is less than 25%. A higher efficiency protocol is needed.

Microchip Inc.'s Method

The latest design including isolated communication in a digital power supply is brought by Microchip Inc. in 2008[33]. The system's diagram is shown in Figure 2-16.

This is a 125 KHz Phase-Shift ZVT Converter. In this system, three opto-couplers are used. Two of them are used for transmitting duty cycles, and the other one is used for transmitting data. As discussed above, transmitting duty cycles is not desirable as the bandwidth limitation of the digital isolator will greatly reduce the duty-cycle resolution. As a result of the limited duty-cycle resolution, ADC resolution must be lowered in order to avoid limit-cycle oscillation. In this system, 8 MSBs of the 10-bit ADC are transmitted to the primary side, thus the output accuracy is 0.39% rather than 0.098% provided by the 10-bit ADC.

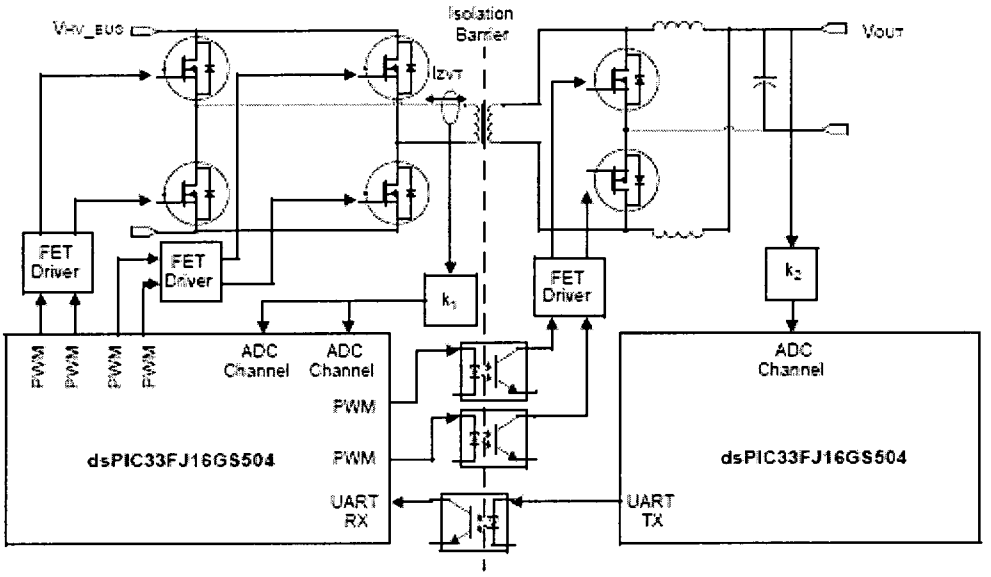


Figure 2-16 Diagram of Microchip's System

The UART (Universal Asynchronous Receiver/Transmitter) standard is used in this system to transmit samples from the secondary side to the primary side. The Baud Rate is configured to 2 Mbps, and each sample is packed into a 12-bit frame: 1 Start bit, 8 data bits, 1 Parity bit, and 2 Stop bits. The efficiency of this protocol is 8/12=66.7% and the overall efficiency of this communication link is 8*125k/2M=50%. While this efficiency is not too bad, UART standard was not designed for real-time communications; increased software and processing effort are required to overcome problems such as clock drift, send jitter, and so on[35].

Chapter 3 The Proposed System

3.1 Introduction

As is discussed in Chapter 2, existing architectures do not support multiplexed use of the digital isolator, provide control functions (e.g. On/Off command, alarms), and all have limited accuracy. The protocols used in existing architectures are not suitable for communication through pulse-transformers. This chapter will introduce a new architecture and protocol design. Section 3.2 introduces the basic idea of the proposed architecture. Section 3.3 discusses considerations in protocol design. Section 3.4 gives the proposed protocol. Section 3.5 introduces the proposed architecture in detail. Section 3.6 compares the proposed system with existing systems and draws a conclusion.

3.2 System Overview

A simplified system block diagram is shown in Figure 3-1 in order to describe the basic idea of the proposed system.

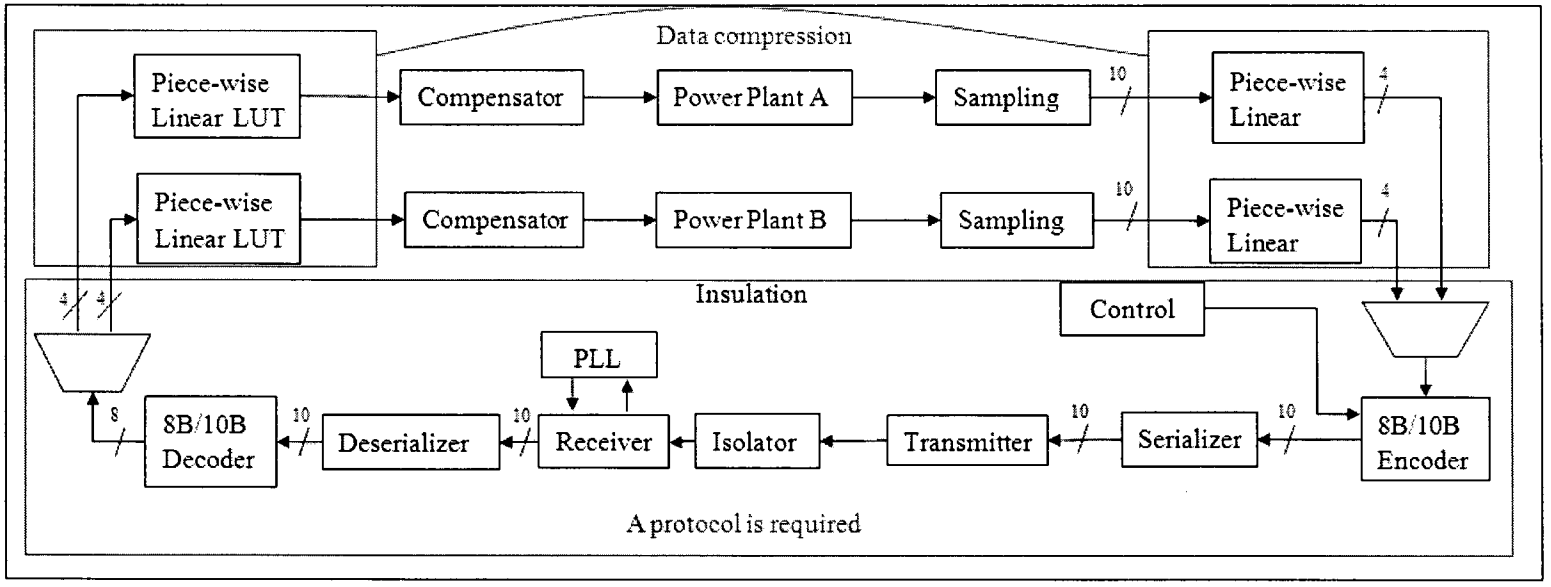


Figure 3-1 The Simplified System Block Diagram

From Figure 3-1 it can be seen that the system includes two power plants. The sampling blocks use 10-bit ADCs to sample the output voltages. The Piece-wise Linear blocks compress 10-bit samples into 4 bits, which will be discussed in Chapter 4. The two power plants transmit data for feedback control through a shared communication bus. Only one isolator is used, and a phase-lock loop (PLL) is used on the primary side to recover the clock. The communication blocks require a carefully designed protocol, which will be discussed in this chapter. As it will be shown in Section 3.5, the system can be expanded to several power stages, all sharing one communication bus to transmit data. The detailed description of the proposed system will be discussed by then.

3.3 Protocol Design Issues

In digital communication, data are typically encapsulated into frames before sending. A frame usually consists of a header, a body, and a footer. The frame header and footer define the beginning and the end of a frame. A time slot may be needed between frames in order to distinguish frame. The header must contain a distinctive bit sequence for frame synchronization. The header may also contain other information such as the frame length, the destination address, and so on. The footer must contain a stop sequence and probably

contains a check bit for error detection. These framing bits, while providing necessary information for communication, reduces code efficiency, therefore should be minimized.

This section discusses the communication issues and proposes a protocol optimized for digital power supplies.

3.3.1 Considerations in Protocol Criteria

This protocol is designed for a multiplexed feedback power system. To design an optimized protocol, following issues must be considered:

A. DC-Balance

Pulse-transformer is a transformer with very small inductance. Its input should be differential input rather than TTL input. Logic 1s can be considered as current flow in one direction, and logic 0s can be considered as current flow in the opposite direction. Within a period of time, the amount of 1s and the amount of 0s transmitted through the pulse-transformer should be equal; otherwise the pulse-transformer will be saturated. To avoid this saturation from happening, a proper line-code must be chosen to provide DC balance within a minimum possible period of time.

Pulse-transformer is a desired candidate of digital isolator because it is significantly cheaper than digital opto-couplers and RF-couplers, and also because it can be used for delivering power while transmitting data [36].

B. Clock Recovery

In order to eliminate a separate isolator for the clock line, clock recovery should be used. The protocol should provide enough state change so that clock recovery is possible. The candidate line-code has to be a self-clocking code.

C. Frame Length

Because the frequency of each power stage can be different, it would be nice if the frame length is variable. For example, in steady state, the error voltage is at most 1-2 LSB away from the reference voltage; if the data length can vary, sometimes only 2 bits are enough to present a sample, and thus the saved bandwidth can be assigned to other channels. The communication will be more flexible and the data bus will be used more efficiently.

However, variable frame length cannot reduce the required bandwidth of the communication bus because the bandwidth must be sufficient for the worst case, where the outputs are not stable every sample will have maximum length.

There are also benefits of using a fixed-length frame.

First of all, the stop sequence can be omitted. The receiver doesn't need to know the end of the frame as long as it knows the beginning and the length.

Second, for a fixed frame length the address bits can be eliminated. The samples can be organized in a certain order within a frame; therefore the address information is already contained in the order.

D. Error Detection

It would be ideal if the frame includes a mechanism for error detection. The power supply control is a real-time system; if a frame is found defective, it must be abandoned, but there is no need to have a mechanism to notify the secondary-side controller to send the same frame again, as the next sample is already generated.

E. Control Symbols

Other than data, we also need some control symbols in the protocol. For example, during the standby state, we need an idle sequence in order to keep the synchronization of the receiver's clock. We also need other control symbols such as Shut-Off, Soft-Start, Over-Current, and so on.

F. Time Slot

Time slot between adjacent frames is not allowed in this protocol. Because the pulse-transformer is driven by a differential input, continuous state of either logic 0 or logic 1 will quickly saturate the pulse-transformer. In order to distinguish frame from frame in the data stream, the start sequence must be distinctive from data bits.

3.3.2 Line Code Comparison

Bearing in mind the protocol criteria discussed in the last Section, we need to choose a line code that meets our requirements.

A number of line codes are available. Some of the line codes are simpler and only provide solutions for one or two criteria, such as DC-balance or Self-clocking. Some other line codes are more sophisticated and provide features for higher level control, such as frame synchronization and other control functions.

In the implementation of digital power supplies, the protocol needs fewer functions than most communication protocols, and because power management chips do not have high computing capacity, the candidate line code should be simple enough for implementing codec circuits.

Let's start with line codes that feature Self-clocking to help Clock Recovery.

(1) Bipolar Return-to-zero (RZ) [37]

RZ code uses a positive pulse to present "1" and a negative pulse to present "0". Between each pulse the signal returns to neutral. For this reason, the RZ code is self-clocking.

It has two major drawbacks:

a. It does not provide DC-balance. For example, if a stream of data is all 1s, the positive current will saturate the pulse-transformer.

b. Its symbol rate is twice as the bit rate – there are two signal transitions in each data bit – therefore it requires twice the bandwidth to send the same number of bits.

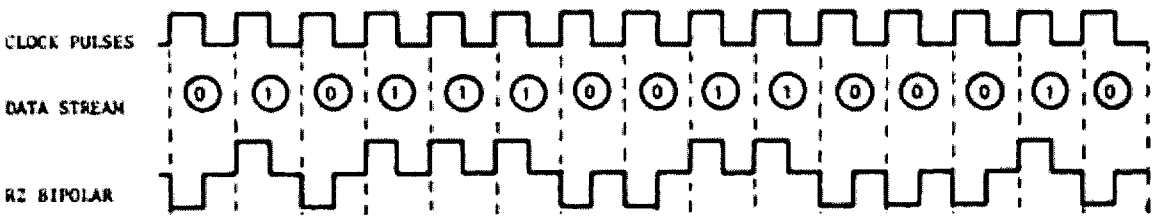


Figure 3-2 Bipolar RZ, from [37]

(2) Manchester code [38][39]

Manchester code sends a "1" as a high-to-low transition, and sends a "0" as a low-to-high transition. It has at least 1 signal transition during each bit period; therefore the clock signal can be recovered from the data signal.

Manchester code inherently solves the DC-balance problem: higher level protocols do not need to address this issue. However, it also requires twice the bandwidth to send the same amount of data.

It does not provide control functions for higher level protocol; therefore frame synchronization must be implemented in a higher level protocol.

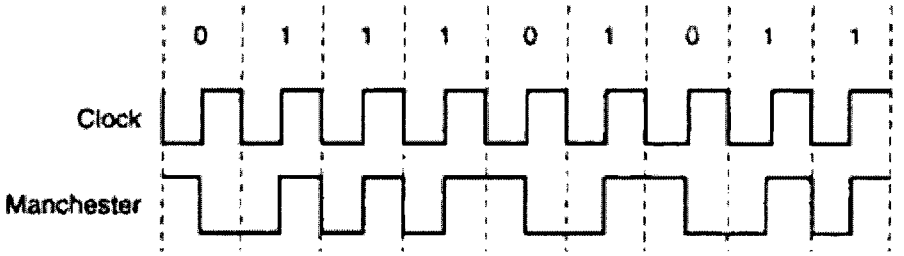


Figure 3-3 Manchester Code, from [39]

(3) 4B/5B [40][39]

4B5B code maps 4-bit words into 5-bit codes in such way that the signal has at least one state transition in each 5B code. 5B codes "00000" and "11111" are usually left unused to avoid clock drift. This helps Clock Recovery because it provides at least one signal transition in every 5 bits. The 5B code provides a collection of 4B words plus some control codes for higher level protocol. Detecting illegal codes (codes not included in the 5B dictionary) is a form of error detection.

Although 4B/5B code provides solutions for Clock Recovery and control words, it does not solve the DC-balance issue. The solution of this issue, as well as frame synchronization, is left to higher level protocol, such as adding a scrambler.

Table 3-1 Data Codes in the 4B/5B Translation Table, from [40]

Hex	Binary	5B Code		Hex	Binary	5B Code
0	0000	11110		8	1000	10010
1	0001	01001		9	1001	10011
2	0010	10100		A	1010	10110
3	0011	10101		B	1011	10111
4	0100	01010		C	1100	11010
5	0101	01011		D	1101	11011
6	0110	01110		E	1110	11100
7	0111	01111		F	1111	11101

(4) 6B/8B [41]

Similar to 4B/5B code, 6B/8B code encodes 6-bit words into 8-bit codes in order to pick out unwanted code patterns and provide useful engineering properties for data communication.

All the 8B codes consist of four 1s and four 0s. This feature guarantees DC-balance in every 8 bits. There are 70 8B codes that include four 1s and four 0s; 64 of them are assigned to the 64 6B words, 4 of them are used as control codes, and the two other codes, 00001111 and 11110000, are left unused in order to prevent clock shift. 6B/8B code also provides enough signal transitions for help Clock Recovery.

One drawback of 6B/8B code is that it does not have a special "delimiter" code that will not be misrecognized in anywhere else that in the data stream. For example, the control code K010101 (01010101) may be misrecognized by 000100 (011**00101**) followed by 110111 (**01010**110). With absence of this feature, the frame synchronization must rely on higher level protocol.

The error detection is performed by detecting illegal codes (codes not included in the 8B dictionary.)

(5) 8B/10B [42][39]

8B/10B code encodes 8-bit words into 10-bit codes to pick out unwanted code patterns and provide useful engineering properties for data communication.

8B/10B provides DC-balance. The mechanism is described below: Each 8B code is mapped into two 10B code. The two 10B codes are reversed codes of each other. The 10B codes are selected such that they are either in the pattern of 5 0s and 5 1s, or in the pattern of 4 0s and 6 1s, or 6 0s and 4 1s. For those 10B codes that consist of 5 0s and 5 1s, their disparities are neutral. For those 10B codes that consist of 4 0s and 6 1s, and their counterpart which consist of 6 0s and 4 1s, their disparities are +2 and -2, respectively. The system's Running Disparity (RD) is monitored by the encoder: At the system's startup, the RD is 0; if the first code to send has neutral disparity, the RD stays unchanged; if the first code to send has non-neutral disparity, the encoder chooses the one that has +2 disparity, and updates the system RD to +2. If the next code sent has neutral disparity, the RD stays at +2; if it has non-neutral disparity, the encoder chooses the one that has -2 disparity to offset the system's +2 RD, and updates the system RD to 0. This way, the DC-balance is achieved within at most 6 bits.

8B/10B code provides enough signal transitions to support Clock Recovery. "The data sequence with the lowest transition density, which can be maintained indefinitely, has an average of 30 transitions per 100-digit interval."[42]

8B/10B code also provides three "comma" codes which can be used for frame synchronization. These codes will not occur anywhere else in a data stream by

two data codes overlapping. These "comma" codes can be used for frame synchronization, emergency shut off, and other functions.

The error detection is performed by detecting illegal codes (codes not included in the 10B dictionary.)

From above comparisons we can see that 8B/10B code is the best line code that meets our criteria. It guarantees DC-balance with less bandwidth than that required by Manchester code; it provides enough transitions of support clock recovery; and it provides distinctive delimiter codes for frame synchronization.

8B/10B code is chosen for designing the proposed protocol.

3.4 The Proposed Protocol

The proposed protocol is designed and optimized for multiplexed feedback power supply implementations. It minimizes the framing bits and other information bits in order to achieve the highest code efficiency possible. By using this protocol, an engineer is able to multiplex as many feedback loops as possible when the total bandwidth is given; or minimize the required bandwidth to save costs when the bandwidth of each feedback loop is given.

The protocol is described as follows:

A. Fixed frame length.

The frame length is fixed, and the number of bytes in each frame is determined by the designer. There is no time slot between frames. The minimum bandwidth of the data bus, $f_{bus,\min}$, is calculated as:

$f_{bus,\min} = 10(N+1)f_{sample,\max}$ where N is the number of data bytes following the "Sync" byte, $f_{sample,\max}$ is the highest sampling frequency (sample/s) of all power stages in the system. The frame frequency f_{frame} equals to $f_{sample,\max}$ to adapt to the fastest loop.

B. Minimized framing header.

The frame header is simply a "Sync" code. It does not contain frame length information or "Stop" bit, because the frame length is fixed.

Each frame carries samples of all channels. The channel information of each sample is indicated in the position order in each frame: The first byte following the "Sync" code is CH1 (or CH1 and CH2, refer to the Piece-wise Linear Feedback method in Chapter 4); the second byte following the "Sync" code is CH2 (or CH3 and CH4), and so on. Therefore the header needs not to describe channel information explicitly.

For a system that consists of only one channel (or two channels with the Piece-wise Linear Feedback method), the "Sync" code is not needed: a frame is simply one byte. In this case, the protocol can achieve its theoretical maximum efficiency. Because each byte is encoded into 8B/10B code, errors can be detected by incorrect 10B patterns and corrected by shifting the frame position.

C. "Sync" codes also indicate frame types.

The sampling frequencies of power stages in a system may vary. The proposed protocol arbitrarily demands each possible frequency in a system be integral times lower than the highest sampling frequency, $f_{sample,\max}$, in order to fit into the frame.

For example, a system consists of 4 power stages with sampling frequencies of: CH1: 100 KHz, CH2: 50 KHz, CH3: 25 KHz, CH4: 25 KHz. A two-byte frame can carry all channels rather than a four-byte frame. Instead, multiple control codes are served as "Sync" codes in order to indicate the frame type:

	Header	Byte 1	Byte 2
Frame Type 1	"Sync 1"	CH1	CH2
Frame Type 2	"Sync 2"	CH1	CH3
Frame Type 3	"Sync 3"	CH1	CH4

Table 3-2 Example Frame Types

Then the frame sequence is below:

Frame Type	Type 1	Type 2	Type 1	Type 3	Type 1	Type 2	Type 1	Type 3	...
Byte 1	CH1	CH1	CH1	CH1	CH1	CH1	CH1	CH1	...
Byte 2	CH2	CH3	CH2	CH4	CH2	CH3	CH2	CH4	...

t

Figure 3-4 Example Frame Sequence

8B/10B code provides 12 control codes; therefore potentially the protocol can support up to 12 frame types. Note that there are only 3 "comma" codes that are distinctive in a data stream. If a frame loses synch because of a missing bit, the following frames will be out of synch until the next "comma" code. For this reason, "comma" codes are recommended in the most frequent frame types.

As will be discussed in the next chapter, a sample can be compressed into 4 bits by Piece-wise Linear Feedback method. Therefore each 8B byte can carry 2 samples, and the system capacity can be doubled!

D. Frames synchronized to PWM

All Switching Mode Power Supplies have ripple voltage on the outputs. The ripple is due to the discrete nature that SMPS delivering power. It is preferable to sample the output voltage at the same point of every cycle; otherwise the output may oscillate at an alias frequency.

3.5 System Architecture and Timing Diagram

As is mentioned in Section 3.2, the system capacity can be extended from two outputs to several outputs, as long as the high-speed digital isolator is fast enough to transmit data. Also, as is discussed in Section 3.3 and 3.4, 8B/10B line code is adopted in this system.

The expanded system block diagram of the proposed system is shown in Figure 3-5. The operation principles are explained below.

In this system block diagram, there are several isolated digital power supplies on the board. Their switching frequencies can be the same, or different by a factor of an integer number, so that switching cycles are synchronized.

10-bit ADCs are used for sampling the output voltages. The sampling is synchronized to the switching cycle. The output voltages are over-sampled by a multiple of the switching frequency in order to eliminate the effect of the voltage ripple.

The secondary-side controller encodes the sample values with a Piece-wise Linear Lookup table (LUT). The LUT converts 10-bit samples into 4-bit Piece-wise Linear codes. This section will be discussed in Chapter 4.

Every two 4-bit Piece-wise Linear codes are combined into an 8-bit code and then encoded into a 10-bit code through an 8B/10B encoder. A Serializer sends 10-bit codes to a transmitter and a pulse-transformer to the primary-side receiver.

Control codes are also 10-bit long and are packed into frames. The frame length depends on the number of power stages and is a fixed length. There is no time slot between frames. If the data line is idle, the "Idle" code should be sent.

A receiver on the primary side receives the 10-bit codes. A Phase Lock Loop is used for clock recovery. The data stream is 8B/10B encoded so that it provides enough state change to help clock recovery.

A Deserializer deserializes the received bits into 10B codes and a Word Aligner is used for recognizing control codes. Once the "Sync" code is recognized, it resets the counter so that the frame following the "Sync" code is synchronized.

The deserialized 10B code is decoded by a 10B/8B decoder. The 8B result is then split into its original two 4-bit Piece-wise Linear codes.

The 4-bit Piece-wise Linear codes are translated back to 10-bit sampled values.

The primary-side controller calculates the next duty cycle based on the received sample values.

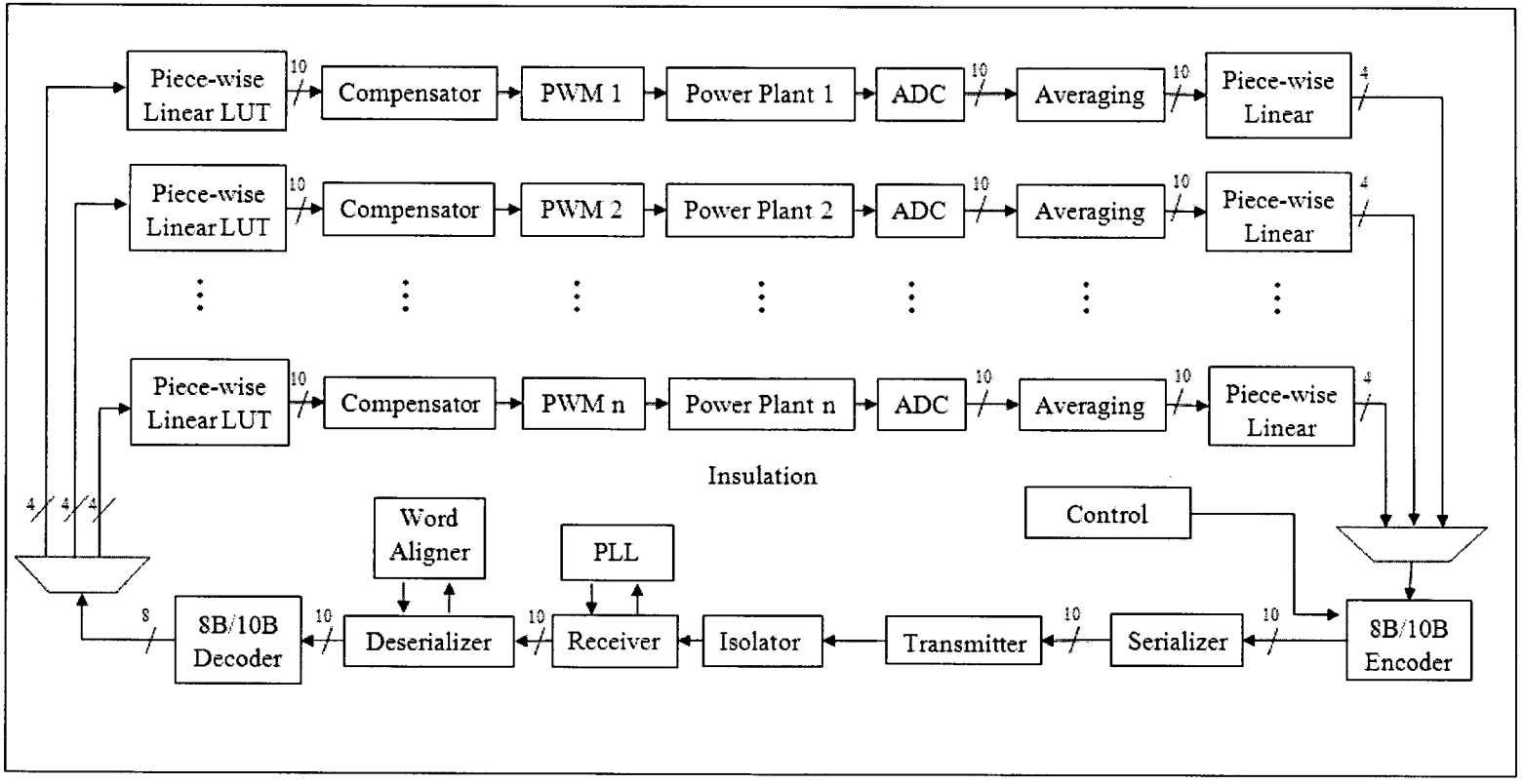


Figure 3-5 Extended System Block Diagram

In order to explain the system's operation principle in detail, an example System Timing Diagram is provided in Figure 3-6. The explanation is as follows:

Assume there are 18 power supplies in a system operating at 100 KHz. The data bus operates at 10 MHz; therefore each frame can carry 9 data bytes plus 1 "Sync" byte. We assign the first data byte to carry PWM1 and PWM2's samples, the second data byte to carry PWM3 and PWM4's samples, and so on.

The PWM1's rising edge simultaneously triggers four ADC conversions of all outputs, and then the averaged sample is encoded by the secondary-side controller. The encoded sample code must be ready while the "Sync" code is in transmission so that the sample code can be sent next to the "Sync" code. The decoder detects the beginning of the frame by recognizing the "Sync" code. It "knows" the samples' addresses by their orders within a frame. It decodes the codes while the next byte is in transmission, and updates the PWMs before the next switching cycle.

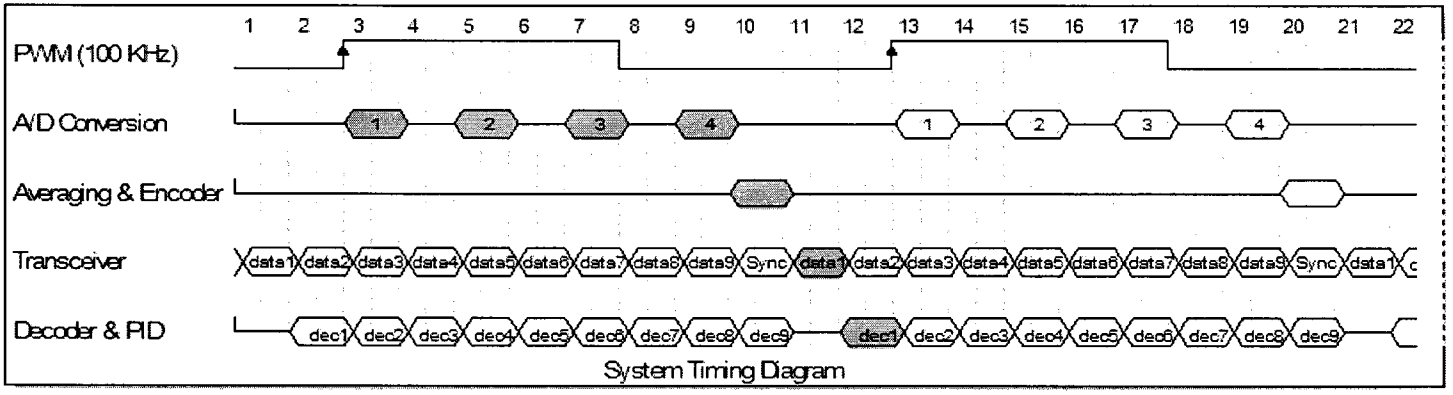


Figure 3-6 System Timing Diagram

3.6 Conclusion

The proposed system provides many advantages comparing to existing methods. Table 3-3 is a comparison of the proposed system and existing methods, which shows the improvements made by this thesis.

Table 3-3 System Evaluation

	The Proposed System	**Ka Leung's System**	**Aleksandar Prodic's System**	**Microchip's System**
Method	Encoded by 8B10B line-code, Piece-wise Linear Feedback method	Directly transmit duty cycles via high speed isolators	Transmit 4 LSB using a customized protocol	Transmit 8 MSB using UART standard
DC-Balance	Yes	No	No	No
Multiplexed bus	Yes	No	No	No
Isolators needed	1 for all multiplexed power supplies	1 for each PWM	2 for each power supply	1 for each power supply
Output Accuracy	0.098% (10-bit ADC)	1.563% (6-bit ADC)	0.392% (8-bit ADC)	0.392% (8-bit ADC)
Control Functions	Yes	No	No	No
Clock Recovery	Support	N/A	Do not support	Do not support
Error Detection	Yes	N/A	Yes	Yes
Protocol Efficiency	5 bits per sample for 2 channels (theoretical maximum) to $5 + \frac{10}{4} = 7.5$ bits per sample for 4 channels (worst case scenario).	N/A	16 bits per sample for 1 channel.	12 bits per sample for 1 channel.

Chapter 4 Piece-wise Linear Feedback Method

4.1 Introduction

The Piece-wise Linear Feedback method (PWL) was previously mentioned in Chapter 3 already. It plays an important role in compressing 10-bit samples into 4 bits while providing better accuracy than existing architectures. Special attention must be paid to understand its impact on dynamic responses. This chapter discusses the design of the PWL and the theoretical analysis. Section 4.2 and 4.3 discuss the theoretical analysis of the Piece-wise Linear Feedback method. Section 4.4 gives a design example. Section 4.5 gives the simulation results.

4.2 Output Dynamic Range and Steady-State Error Range

The guideline of compressing data is to eliminate useless and unnecessary information.

Assuming in a 5V output power supply, a 10-bit ADC provides 0.2% resolution from 0V to 10V. If according to the design specification, the output voltage should never be over 5.5V or lower than 4.5V, then the ADC range 0V-4.49V and 5.51V-10V are useless information. Similarly, if the design objective is to keep the output voltage within 5V±0.01V, the resolution of ±0.01V at 4.5V is unnecessary information. Therefore, the

sample data can be compressed by eliminating useless voltage range and unnecessary resolution.

Output Dynamic Range

The output dynamic range is usually defined in the design specification by the users. Some power supplies are very tight on this range, which may be about ±1% variation for a step load transient. Some design specifications are less tight. A variation of ±5% range is a fair estimation. Dynamic range varies from design to design.

Steady-State Error Range

In addition to eliminating unused voltage range, we can further compress the data by reducing unnecessary resolution.

The control objective of a power supply is to stabilize the output voltage at a rated value with very small variation. The highest resolution should be provided at ± a few LSBs of the rated voltage value. The further away from the rated value, the less resolution is needed, because the feedback system will bring the voltage back to the desired point.

Before we study how the reduced resolution impacts on the system's dynamic response, we need to understand that in which range the resolution can NOT be reduced. The answer to this question depends on the compensator design:

In a typical Phase-Lead or Phase-Lag compensation which do not have integral component in the system (Type 0), a steady-state error is introduced into the output voltage.

For example, a 5V output power supply may drop to 4.99V for 10% load; and the output voltage may further drop to 4.95V for 100% load: With no integral control, when the load increases, the output voltage drops a small value to increase V_{error} The V_{error} is multiplied by the compensation gain and increase the PWM duty cycle to deliver more power to the load.

Within a load range, the operation point of a Type 0 power supply moves within the steady-state error range. For example, for a 5V power supply whose output drops to 4.95V at full load, the operation point moves from 5V±1LSB to 4.95V±1LSB. Within 5V to 4.95V, any point could be the operation point depending on the load. Therefore, the highest resolution must be available throughout this range.

One way of calculating the steady-state error of a Type 0 system is to derive the system transfer function (see Section 4.4.2), and then use the formula below, from [43]

$$e_{ss} = \frac{A}{1+K_p} \tag{4.1}$$

where e_{ss} is the steady-state error, A is the magnitude of a step input, Kp is called position error constant, which is the DC gain of a Type 0 system.

From data compression point of view, we hope the steady-state error is as small as possible, because a large steady-state error power supply needs more resolution information than a small steady-state error power supply. However, there is an engineering trade off:

In order to reduce e_{ss}, we need to increase the system DC gain. There are two ways to increase the system DC gain: position a pole near 10 Hz or lower, or increase the total gain of the compensator. The first method significantly reduces the loop bandwidth, but is sometimes infeasible due to other constraints such as phase margin. The second method shifts the entire system gain curve up, which is also constricted by the stability criteria.

PID compensation is one way to solve this contradiction. By introducing an integral component into the controller, the steady-state error can be eliminated. The highest resolution range is limited within a few LSBs around the nominal voltage, thus the sample data are compressed as much as possible.

4.3 Piece-wise Linear Feedback Method

The basic idea of the Piece-wise Linear Feedback method is to develop a Lookup table such that several adjacent sample values are mapped into the nearest table values in order to reduce the resolution for large sample errors. The design objective is to use 4 bits to cover the output dynamic range, because two 4-bit samples can compose an 8-bit byte for encoding by 8B/10B line-code.

The resultant resolution distribution of a 4-bit sample is in piece-wise linear manner, therefore this method is called Piece-wise Linear Feedback method.

The reduced resolution effectively adds extra gain into the system's transfer function. This fact can be explained as follows:

Assume $V_{error} = 0.05V$. Due to the reduced resolution, if the nearest discrete sample value is 0.06V, then 0.06V will be feedback, and the equivalent extra gain is 0.06/0.05=1.2. Likewise, if the nearest discrete sample value is 0.04V, then the equivalent extra gain is 0.04/0.05=0.8.

With the understanding of the extra gain, it is easy to develop design guidelines:

Guideline 1: The extra gain remains the same if we assign high resolution at small V_{error} values and low resolution at large V_{error} values; (see Figure 4-8).

Explanation: Consider $V_{error} = 0.04V$ and the nearest sample value is 0.05V, the equivalent extra gain is 1.25. Consider $V_{error} = 0.4V$ and the nearest sample value is 0.5V, the equivalent extra gain is also 1.25, while the resolution at 0.4V is much lower than which at 0.04V.

Guideline 2: Keep the extra gain within a safe range so that it does not cause the system instability, especially at small V_{error} values.

Explanation: The extra gain is changing and unpredictable during dynamic transitions. It will add disturbance on the system's dynamic response. Before we design such a Piece-wise Linear Lookup table, we need to look at the system's Bode plot to make sure that this amount of extra will not cause the system unstable. Especially when V_{error} is small, the

output voltage is very close to the nominated value, an improper extra gain may result in wrongful compensation and the system will be unstable.

An example of the Piece-wise Linear Feedback method is shown in Figure 4-1. The 16 4-bit codes except 0 are distributed into 3 linear areas: codes 1~5 evenly cover the 5% ~ 0.5% error range; codes 6~10 are centered at 0.38% ~ -0.38% range to provide the highest resolution; codes11~15 evenly cover the -0.5%~ -5% range; code 0 stands for under voltage. This way, only 4 bits are enough to cover the ±5% Output Dynamic Range while providing necessary accuracy at steady state.

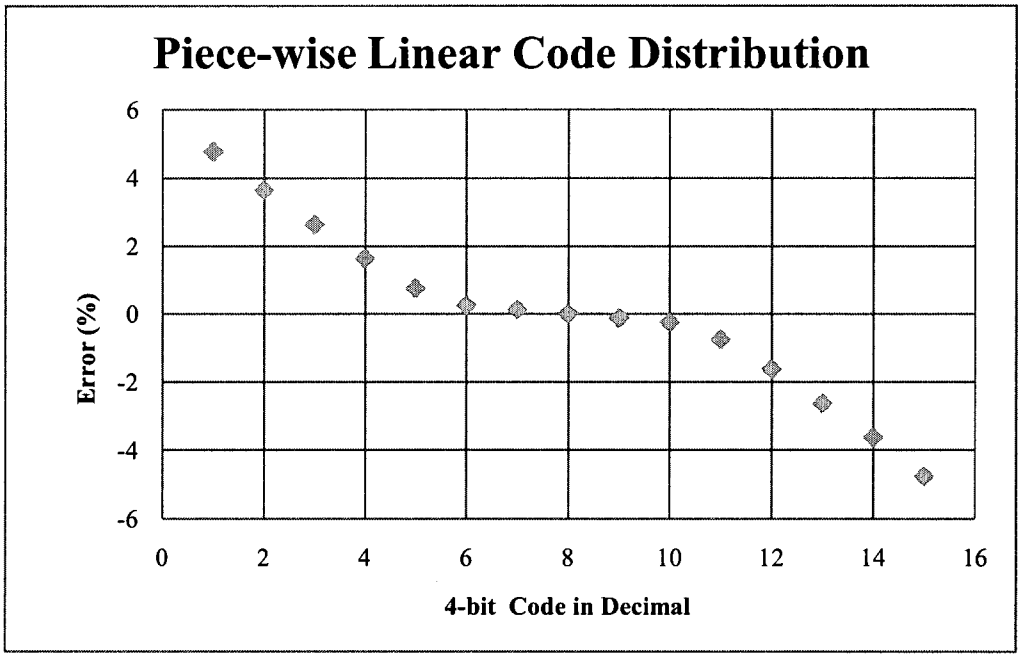


Figure 4-1 Piece-wise Linear Levels Distribution

The impact of Piece-wise Linear Feedback method on a dynamic system is difficult to interpret by equations or Bode plot. The best way to examine this method is by computer simulation.

The following sections will give a design example of the Piece-wise Linear method and the simulation results.

4.4 Design Example

4.4.1 Power Plant Design

Design Specification

Vo	5V
Po (max)	25W
Io (max)	5A
Io (min)	1A
Vi (max)	57V
Vi (min)	36V
Switching Frequency	100 KHz

Design Calculation

A power plant is designed below for evaluating the Piece-wise Linear Feedback method. The design procedure is from reference [15].

Turns ratio: $\frac{N_p}{N_s} = 3$ as stated in the previous text.

Maximum on time: $T_{on,\max} = \frac{(V_o+1)(N_p/N_s)(0.8Ts)}{(V_{in,\min}-1)+(V_o+1)N_p/N_s} = \frac{6\times3\times0.8\times10\mu}{(36-1)+6\times3} = 2.717\mu$

Magnetic: $L_m = \frac{(V_{in,\min}\cdot T_{on,\max})^2}{2.5\cdot T_s\cdot P_o} = \frac{(36\times2.717\times10^{-6})^2}{2.5\times10\times10^{-6}\times25} = 15.31\mu H$

Primary peak current: $I_p = \frac{V_{in,\min}\cdot T_{on,\max}}{L_m} = \frac{36\times2.717\mu}{15.31\mu} = 6.39A$

Primary RMS current: $I_{rms(primary)} = \frac{I_p}{\sqrt{3}}\cdot\sqrt{\frac{T_{on,\max}}{T_s}} = \frac{6.39}{\sqrt{3}}\cdot\sqrt{\frac{2.717\mu}{10\mu}} = 1.923A$

Secondary reset time: $T_r = 0.8T_s - T_{on,\max} = 8\mu - 2.717\mu = 5.283\mu$

Secondary RMS current: $I_{rms(secondary)} = \frac{I_p(N_p/N_s)}{\sqrt{3}}\cdot\sqrt{\frac{T_r}{T_s}} = \frac{6.39\times3}{\sqrt{3}}\cdot\sqrt{\frac{5.28.\mu}{10\mu}} = 8.04A$

Output capacitor (chosen by Bode plot design): $C_o = 1450\mu F$

4.4.2 Establishment of Small-Signal Model of Discontinuous-Mode Flyback Converters

The Flyback power stage can be seen as a current source. It operates in the discrete domain at a frequency much higher than the dynamic response bandwidth. Its transfer function can be derived by exploring the mechanism of the energy transformation of each switching cycle. The output filter is considered continuous because its bandwidth is significantly lower than the switching frequency, therefore its transfer function can be derived in S-domain directly.

Power supplies work at a DC level, which is set by the reference voltage and load. The dynamic response in frequency domain is a small-signal model at a certain DC point. Eliminating the DC component and assuming the input voltage is stable, the closed control loop is shown in Figure 4-2.

As we will see later, the power stage's transfer function, from ΔV_{ea}, the output signal of the compensator, to ΔI_{ave}, the average output current of the power stage, is a pure gain. The power stage's discrete transfer function can be directly integrated with the S-domain transfer function of the output filter.

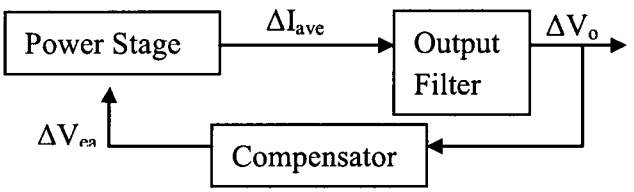


Figure 4-2 Control Loop Chart

The derivation of the small-signal model is below.

$I_p(k)$ – Primary side peak current

$I_s(k)$ – Secondary side peak current

V_{in}– Input voltage

$V_o(k)$ – Output voltage

$V_{ea}(k)$ – Output of the compensator

V_{saw} – Voltage of the sawtooth waveform in PWM

T_s – Sampling period and switching period

$t_r(k)$ – Reset time of the secondary winding which discharges the energy

N – Turns ratio Np/Ns

L_m – Transformer magnetic inductance

C_o – Output capacitor

R_c – ESR of the output capacitor

R_o – Resistive Load

1. The input signal is $V_{ea}(k)$, thus the primary peak current is:

$$I_p(k) = \frac{V_{in} \cdot V_{ea}(k) \cdot T_s}{L_m \cdot V_{saw}} \tag{4.2}$$

2. The secondary peak current is:

$$I_s(k) = N \cdot I_p(k) \tag{4.3}$$

3. Consider the primary winding, the Volt-second of charging time should equal to the Volt-second of the discharging time in each cycle, therefore the reset time in which the secondary current decays from $I_s(k)$ to zero is:

$$t_r(k) = \frac{V_{in} \cdot V_{ea}(k) \cdot T_s}{V_o(k) \cdot V_{saw} \cdot N} \tag{4.4}$$

4. The secondary winding sources current during $t_r(k)$. The average current of each

switching cycle can be calculated by multiplying the average current of the reset time, which is $\frac{I_s(k)}{2}$, by the reset time $t_r(k)$, and dividing by the switching period T_s:

$$I_{ave}(k) = \frac{I_s(k) \cdot t_r(k)}{2T_s} \tag{4.5}$$

5. Plug Equation (4.2), (4.3), and (4.4) into (4.5), we get:

$$\begin{aligned} & I_{ave}(k) \\ &= \frac{V_{in} \cdot V_{ea}(k) \cdot T_s \cdot N}{2 \cdot T_s \cdot V_{saw} \cdot L_m} \cdot \frac{V_{in} \cdot V_{ea}(k) \cdot T_s}{V_o(k) \cdot V_{saw} \cdot N} \\ &= \frac{V_{in}^2 \cdot V_{ea}{}^2(k) \cdot T_s}{2 \cdot L_m \cdot V_{saw}{}^2 \cdot V_o(k)} \end{aligned} \tag{4.6}$$

6. Small-signal model:

The small-signal model is only correct at a given DC level. Writing each variable in a form of the sum of a DC component and an AC component, the small-signal model of a Flyback power stage can be derived as follow:

$$\begin{aligned} & I_{ave}(k) = \frac{V_{in}^2 \cdot V_{ea}{}^2(k) \cdot T_s}{2 \cdot L_m \cdot V_{saw}{}^2 \cdot V_o(k)} \\ & \Rightarrow I_{ave} + \Delta I_{ave}(k) = \frac{V_{in}^2 \cdot [V_{ea} + \Delta V_{ea}(k)]^2 \cdot T_s}{2 \cdot L_m \cdot V_{saw}{}^2 \cdot [V_o + \Delta V_o(k)]} \\ & \Rightarrow I_{ave} \cdot 2 \cdot L_m \cdot V_{saw}{}^2 \cdot [V_o + \Delta V_o(k)] \\ & + \Delta I_{ave}(k) \cdot 2 \cdot L_m \cdot V_{saw}{}^2 \cdot [V_o + \Delta V_o(k)] = V_{in}^2 \cdot [V_{ea} + \Delta V_{ea}(k)]^2 \cdot T_s \\ & \Rightarrow I_{ave} \cdot 2 \cdot L_m \cdot V_{saw}{}^2 \cdot V_o + I_{ave} \cdot 2 \cdot L_m \cdot V_{saw}{}^2 \cdot \Delta V_o(k) \\ & + \Delta I_{ave}(k) \cdot 2 \cdot L_m \cdot V_{saw}{}^2 \cdot V_o + \Delta I_{ave}(k) \cdot 2 \cdot L_m \cdot V_{saw}{}^2 \cdot V_o \cdot \Delta V_o(k) \\ & = V_{in}^2 \cdot T_s \cdot V_{ea}{}^2 + V_{in}^2 \cdot T_s \cdot \Delta V_{ea}(k)^2 + V_{in}^2 \cdot T_s \cdot 2 \cdot V_{ea} \cdot \Delta V_{ea}(k) \end{aligned} \tag{4.7}$$

Eliminating the DC components and 2nd order AC components, we get:

$$I_{ave}\cdot 2\cdot L_m\cdot V_{saw}{}^2\cdot \Delta V_o(k)+\Delta I_{ave}(k)\cdot 2\cdot L_m\cdot V_{saw}{}^2\cdot V_o = V_{in}^2\cdot T_s\cdot 2\cdot V_{ea}\cdot \Delta V_{ea}(k)$$

$$\Rightarrow \Delta I_{ave}(k)=\frac{V_{in}^2\cdot T_s\cdot 2\cdot V_{ea}\cdot \Delta V_{ea}(k)}{2\cdot L_m\cdot V_{saw}{}^2\cdot V_o}-\frac{I_{ave}\cdot 2\cdot L_m\cdot V_{saw}{}^2\cdot \Delta V_o(k)}{2\cdot L_m\cdot V_{saw}{}^2\cdot V_o}$$

$$\Rightarrow \Delta I_{ave}(k)=\frac{V_{in}^2\cdot T_s\cdot V_{ea}\cdot \Delta V_{ea}(k)}{L_m\cdot V_{saw}{}^2\cdot V_o}-\frac{I_{ave}\cdot \Delta V_o(k)}{V_o} \tag{4.8}$$

Note that $\Delta V_o(k)=\Delta I_{ave}(k)\cdot R_o$ and $I_{ave}\cdot R_o = V_o$

$$\Delta I_{ave}(k)=\frac{V_{in}^2\cdot T_s\cdot V_{ea}\cdot \Delta V_{ea}(k)}{L_m\cdot V_{saw}{}^2\cdot V_o}-\frac{I_{ave}\cdot \Delta V_o(k)}{V_o}$$

$$\Rightarrow \Delta I_{ave}(k)=\frac{V_{in}^2\cdot T_s\cdot V_{ea}\cdot \Delta V_{ea}(k)}{L_m\cdot V_{saw}{}^2\cdot V_o}-\frac{I_{ave}\cdot \Delta I_{ave}(k)\cdot R_o}{V_o}$$

$$\Rightarrow \Delta I_{ave}(k)\cdot(1+\frac{I_{ave}\cdot R_o}{V_o})=\frac{V_{in}^2\cdot T_s\cdot V_{ea}\cdot \Delta V_{ea}(k)}{L_m\cdot V_{saw}{}^2\cdot V_o}$$

$$\Rightarrow \frac{\Delta I_{ave}(k)}{\Delta V_{ea}(k)}=\frac{V_{in}^2\cdot T_s\cdot V_{ea}}{2\cdot L_m\cdot V_{saw}{}^2\cdot V_o} \tag{4.9}$$

Using the relation that the output power $T_s\cdot V_o^2/R_o$ equals to the input power $L_m\cdot I_p^2/2$, we have

$$T_s\cdot V_o^2/R_o = L_m\cdot I_p^2/2 \Rightarrow T_s\cdot V_o^2/R_o = L_m\cdot(\frac{V_{in}\cdot V_{ea}\cdot T_s}{L_m\cdot V_{saw}})^2\Big/2$$

$$\Rightarrow V_{ea}=\frac{\sqrt{\frac{T_s\cdot V_o^2}{R_o}\cdot\frac{2}{L_m}}\cdot L_m\cdot V_{saw}}{V_{in}\cdot T_s}=\frac{V_o\cdot V_{saw}}{V_{in}}\sqrt{\frac{2\cdot L_m}{R_o\cdot T_s}} \tag{4.10}$$

Plugging (4.10) into (4.9), we get the small-signal model transfer function

$$\frac{\Delta I_{ave}(k)}{\Delta V_{ea}(k)}=\frac{V_{in}^2\cdot T_s}{2\cdot L_m\cdot V_{saw}{}^2\cdot V_o}\cdot\frac{V_o\cdot V_{saw}}{V_{in}}\sqrt{\frac{2\cdot L_m}{R_o\cdot T_s}}=\frac{V_{in}}{V_{saw}}\sqrt{\frac{T_s}{2\cdot L_m\cdot R_o}} \tag{4.11}$$

Assume the overall efficiency is Eff . Reflect the power loss to the input power, we get:

$$P_{effective} = Eff \cdot P_{in} \Rightarrow \frac{V_{effective}^2}{R_{in}} = \frac{Eff \cdot V_{in}^2}{R_{in}} \Rightarrow V_{effective} = \sqrt{Eff}\, V_{in} \qquad (4.12)$$

where $R_{in} = V_{in}/I_{in}$, the input resistance of a power supply and is a negative value.

Substitute V_{in} by $\sqrt{Eff}\, V_{in}$ in (4.11), and we get the final power stage transfer function:

$$\frac{\Delta I_{ave}(k)}{\Delta V_{ea}(k)} = \frac{\sqrt{Eff}\, V_{in}}{V_{saw}} \sqrt{\frac{T_s}{2 \cdot L_m \cdot R_o}} = \frac{V_{in}}{V_{saw}} \sqrt{\frac{Eff \cdot T_s}{2 \cdot L_m \cdot R_o}} \qquad (4.13)$$

This small-signal model has been validated by PSIM simulation.

7. Transfer function of the output filter:

The equivalent circuit of the output filter network is in Figure 4-3.

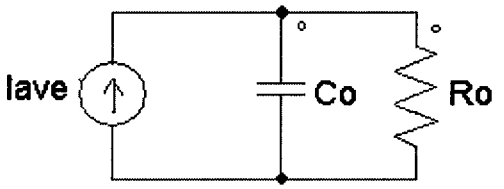


Figure 4-3 Output Filter Circuit

Its Thévenin equivalent circuit is in Figure 4-4.

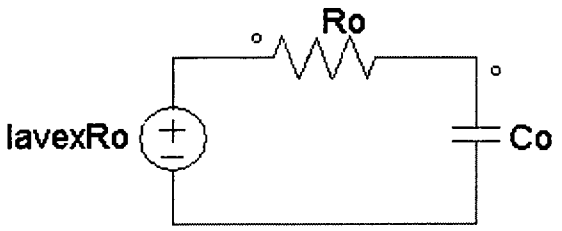


Figure 4-4 Thévenin Equivalent Circuit of the Output Filter

The output voltage V_o is the voltage across the C_o. Therefore the transfer function of the output filter is:

$$\frac{\Delta V_o}{\Delta I_{ave} R_o} = \frac{Z_{C_o}}{Z_{C_o} + R_o} = \frac{\frac{1}{sC_o} + R_c}{\frac{1}{sC_o} + R_c + R_o} = \frac{R_c(s + \frac{1}{R_c C_o})}{(R_o + R_c)(s + \frac{1}{C_o(R_o + R_c)})}$$

$$\Rightarrow \frac{\Delta V_o}{\Delta I_{ave}} = \frac{R_o R_c(s + \frac{1}{R_c C_o})}{(R_o + R_c)(s + \frac{1}{C_o(R_o + R_c)})} \tag{4.14}$$

where R_c is the ESR of C_o.

8. Combining the power stage's transfer function and the output filter's transfer function, we get the total transfer function without the compensator:

$$\frac{\Delta V_o}{\Delta V_{ea}} = \frac{\Delta I_{ave}}{\Delta V_{ea}} \cdot \frac{\Delta V_o}{\Delta I_{ave}} = \frac{V_{in}}{V_{saw}} \sqrt{\frac{Eff \cdot T_s R_o}{2 \cdot L_m}} \cdot \frac{R_c(s + \frac{1}{R_c C_o})}{(R_o + R_c)(s + \frac{1}{C_o(R_o +} \tag{4.15}$$

9. Use the MATLAB function Gz=c2d(Gs, Ts, 'zoh') to convert the S-domain transfer

function into Z-domain.

4.4.3 PID Controller Design

Plug in the design values into the transfer function that is derived in Section 4.4.2, and then use MATLAB to draw the Bode plot. The power plant's frequency response is shown in Figure 4-5.

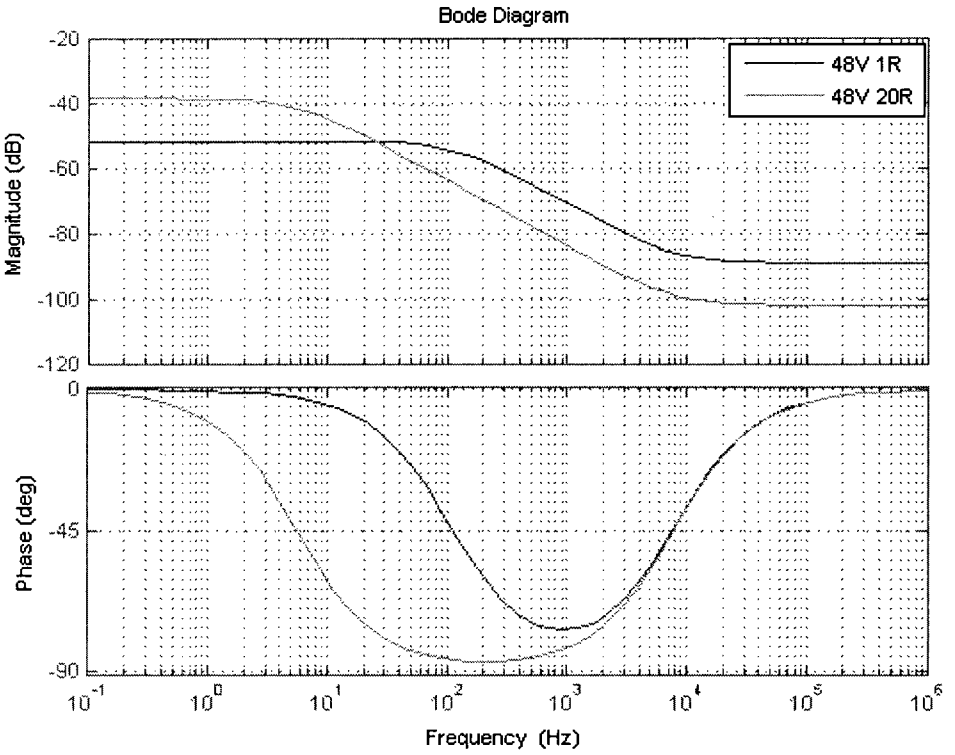


Figure 4-5 Power Plant Bode Plot

From Figure 4-5 we can observe that there is a pole that moves with load, which can be also observed from the transfer function. The controller must be able to compensate both scenarios.

The designed PID controller is

$$D = K_p + K_i \frac{1}{1-z^{-1}} + K_d(1-z^{-1}) \tag{4.16}$$

where $K_p = 30$, $K_i = 0.02$, $K_d = 5$, and z is the z-transform variable.

These PID coefficients were tuned manually and subject to change in the real circuit, as the real circuit's transfer function will differ from what was derived mathematically, due to the non-ideal components and layout. Note that an ADC quantization gain has been taken into account when designing the PID controller. The ADC quantization gain is:

$$\frac{1LSB}{V_{LSB}} = \frac{1LSB}{0.00627\text{V}} = 159.5 \tag{4.17}$$

The basic principle of designing a PID controller is to ensure sufficient phase margin and gain margin; the DC gain should be larger than 50dB to provide fast response and to eliminate steady-state error. Note that with PID control, the phase converges to 0° at the Nyquist frequency, and the system seems always stable regardless the gain magnitude. However, in reality the system can oscillate if the gain is not properly attenuated at higher frequencies. A possible explanation is that when it approaches the Nyquist frequency, the mathematical approximation no longer stands, thus the equations for calculating Bode Plots are no longer valid. Also note that with PID control, the system's gain does not roll off at high frequency; therefore attention must be paid to ensure sufficient attenuation at high frequency in order to avoid oscillation.

The compensated system Bode plot is shown in Figure 4-6. We can see that the DC gains ramp up to infinite, which should provide zero steady-state error so that the high-resolution voltage range can be minimized in Piece-wise Linear Lookup table. We can also see that the compensated system provides around 30dB gain margin (assuming the phase roll off across 180° somewhere between 20 KHz and 100 KHz), and around 90° phase margin, which is sufficient for a stable system [17]. This design provides enough margins for the equivalent extra gain that is introduced by the Piece-wise Linear Feedback method.

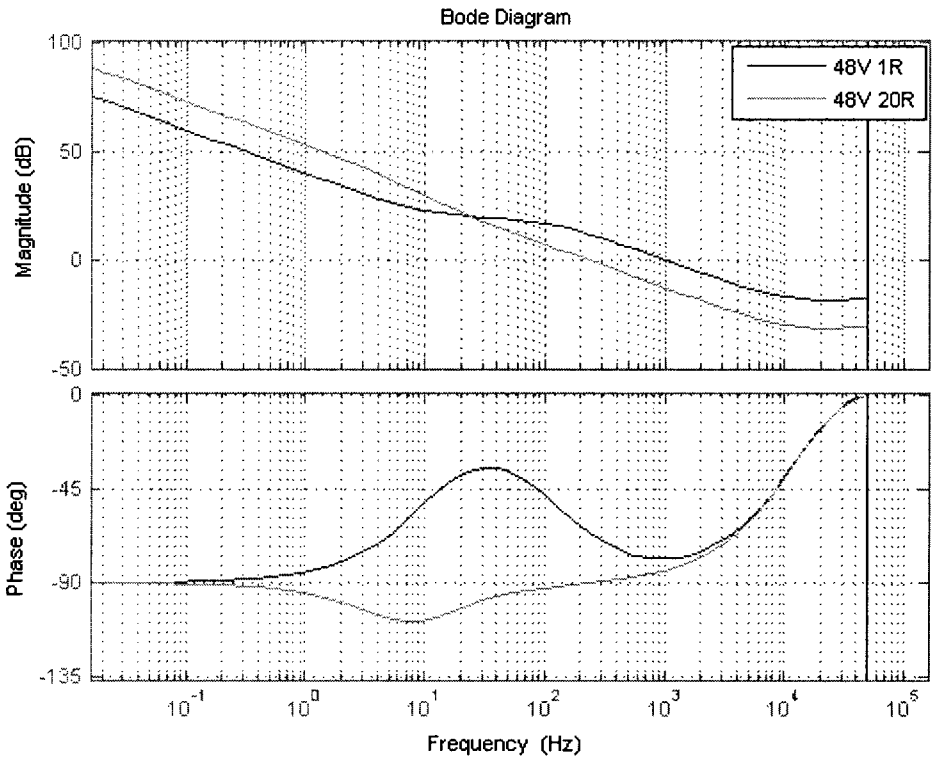


Figure 4-6 Compensated Loop Bode Plot

4.4.4 Piece-wise Linear Lookup table Design

The Bode plot confirmed that the steady-state error is minimized by the PID controller and that the system has sufficient margin to tolerant extra gain before oscillation. Now we have the confidence to safely develop a Piece-wise Linear (PWL) Lookup table.

Following the design guidelines discussed in Section 4.3, an example Piece-wise Linear Lookup table is shown in Table 4-1. The calculation is based on an assumption that a 10-bit ADC is used in this example and its quantization range is from 0V to 6.42V. The same PWL LUT will be used in both PSIM simulation and hardware implementation.

Table 4-1 Example Piece-wise Linear Lookup table

Output Voltage(V)	Verror (V)	Verror (LSB)	PWL Error (LSB)	PWL Error (V)	4-bit Code
5.2649 to 5.2147	-0.2632 to -0.2131	-42 to -34	-38	-0.2382	1111
5.2085 to 5.1583	-0.2068 to -0.1567	-33 to -25	-29	-0.1818	1110
5.1521 to 5.1082	-0.1504 to -0.1066	-24 to -17	-21	-0.1316	1101
5.1019 to 5.0580	-0.1003 to -0.0564	-16 to -9	-13	-0.0815	1100
5.0518 to 5.0267	-0.0501 to -0.0251	-8 to -4	-6	-0.0376	1011
5.0204 to 5.0142	-0.0188 to -0.0125	-3 to -2	-2	-0.0125	1010
5.0079	-0.0063	-1	-1	-0.0063	1001
5.0016	0	0	0	0.0000	1000
4.9954	0.0063	1	1	0.0063	0111
4.9891 to 4.9828	0.0125 to 0.0188	2 to 3	2	0.0125	0110
4.9766 to 4.9515	0.0251 to 0.0501	4 to 8	6	0.0376	0101
4.9452 to 4.9013	0.0564 to 0.1003	9 to 16	13	0.0815	0100
4.8951 to 4.8512	0.1066 to 0.1504	17 to 24	21	0.1316	0011
4.8449 to 4.7948	0.1567 to 0.2068	25 to 33	29	0.1818	0010
4.7885 to 4.7384	0.2131 to 0.2632	34 to 42	38	0.2382	0001

Figure 4-7 shows the distribution of the Piece-wise Linear levels. The 16 codes except 0 are distributed into 3 linear areas: codes 1~5 evenly cover the 0.25V~0.025V range; codes 6~10 are centered at 0.02V ~ -0.02V range to provide the highest resolution; codes11~15 evenly cover the -0.025~ -0.25V range; code 0 stands for under voltage.

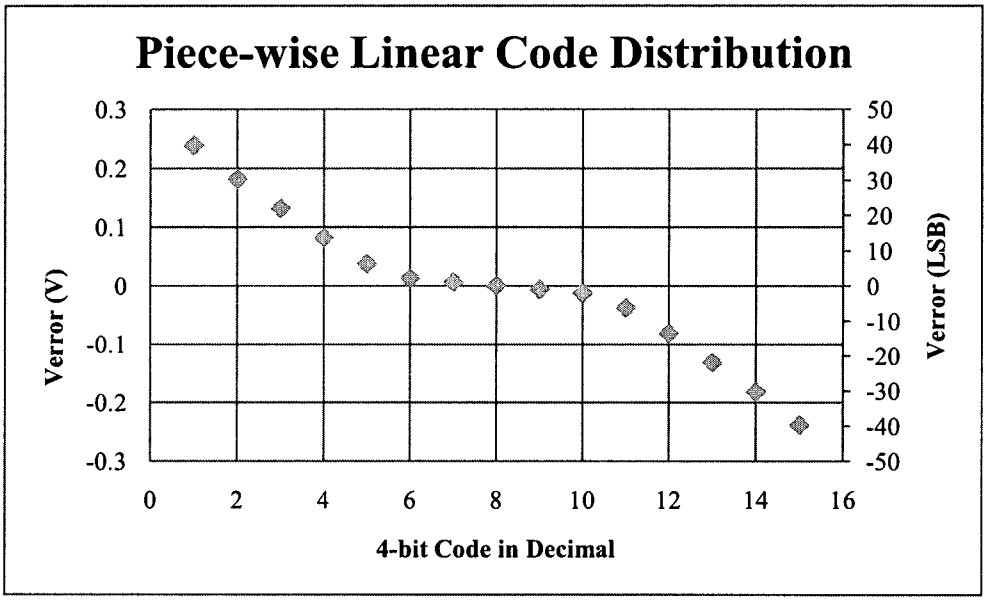


Figure 4-7 Piece-wise Linear Code Distribution

The extra gain introduced by the Piece-wise Linear LUT is plotted in Figure 4-8.

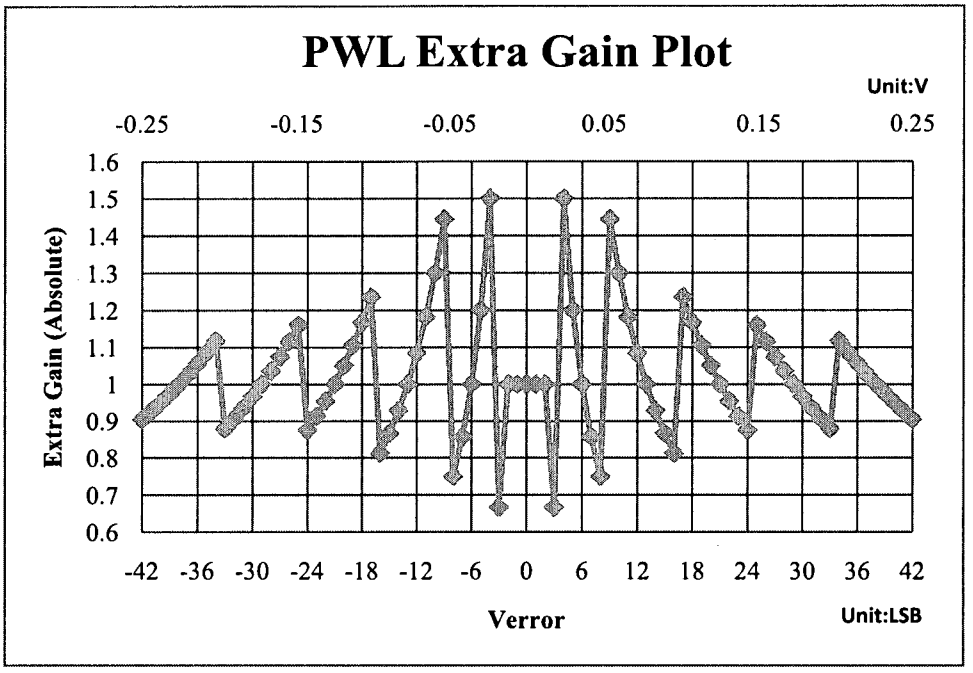


Figure 4-8 Extra Gain Plot

One can see that in the central area the extra gain is (or very close to) 1 and provides accurate error voltage feedback. The extra gains that are adjacent to the central area are the largest in the entire piece-wise linear range, because the Verror in this area is very small and very sensitive to error. Nevertheless, through the entire output dynamic range, the extra gain is controlled less than 1.5; therefore, the Piece-wise Linear Feedback method should virtually have no influence on the system stability.

4.5 Simulation Results

A circuit-based modeling software PSIM is used for examining the proposed Piece-wise Linear Feedback method. The circuit model is shown in Figure 4-9.

The power train is colored with red, and the control circuit is colored with green. Vo is the output voltage. The Quantization block simulates a 10-bit ADC; it over samples the output voltage, and then calculates the average value during a switching cycle in order to filter out the ripple. Subtract the Digital_sample, whose unit is LSB, from a reference value, Digital_ref, and get the error, Digital_error. PWL_LUT is the Piece-wise Linear Lookup table, and its output PWL_error is the piece-wise linear encoded error. H(z) is the PID control law. DPWM_saw is the sawtooth waveform of the DPWM, which is virtually an up counter and constantly compares its value with the compensated error from H(z).

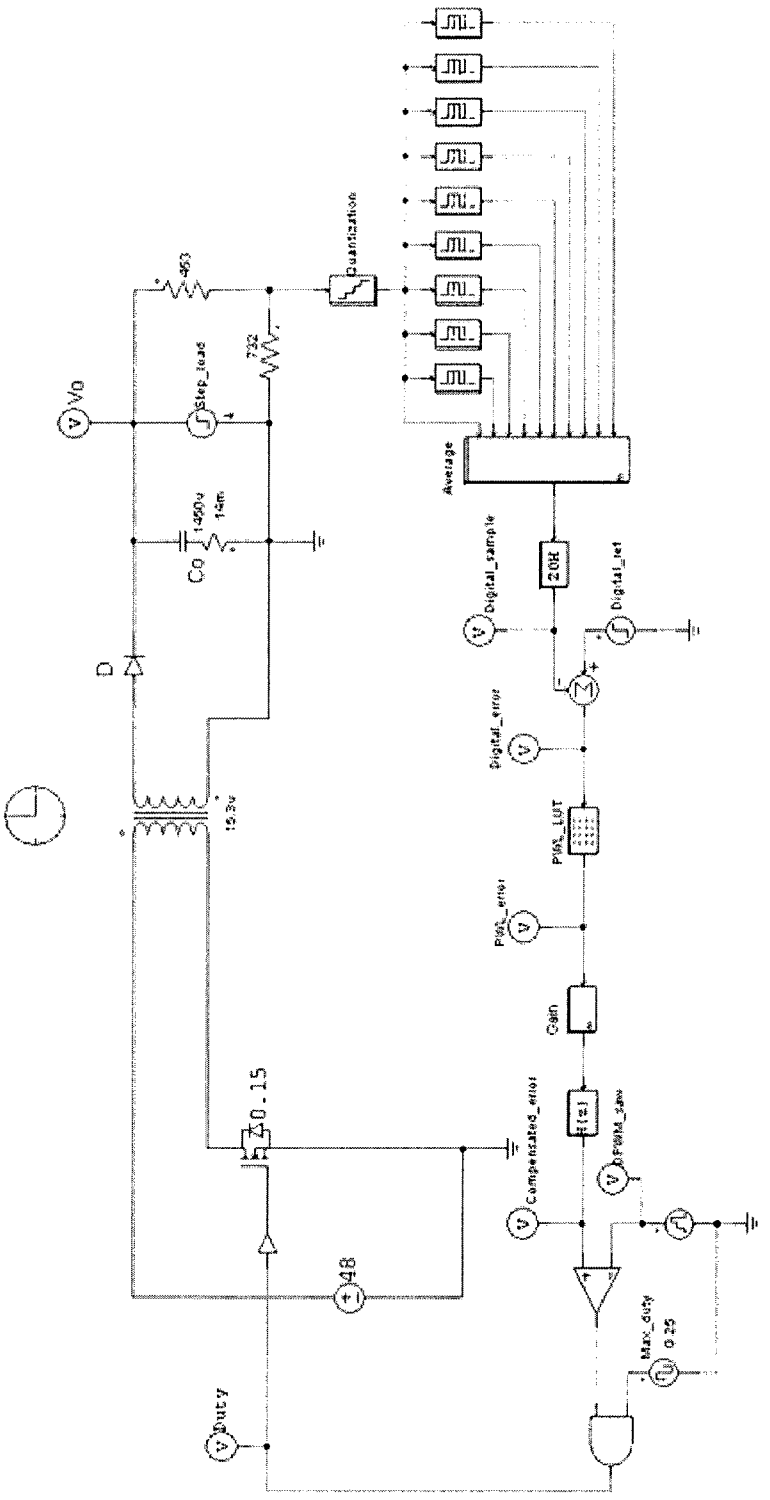


Figure 4-9 PSIM Model

This simulation is for modeling the Piece-wise Linear Feedback method only. It does not include the communication protocol proposed in Chapter 3.

The signals of interest include: the output voltage and error signal. The simulation runs with, or without, the Piece-wise Linear Lookup table, and a comparison of these signals reveal the impact of the PWL LUT on the dynamic response.

Figure 4-10 shows the system's response to a step load transient from 25% to 75%. The two waveforms on the upper side are from the proposed piece-wise linear encoded system; the two waveforms on the lower side are from the system without the PWL LUT. PWL_error is the error signal encoded by the PWL LUT. Digital_error is the original error signal. The locations of these signals can be found in Figure 4-9.

From Figure 4-10 one can see that, although the PWL_error waveform appears obvious piece-wise linear characteristics – coarse resolution at large error – the output voltage's waveforms of the two systems are almost identical.

Figure 4-11 shows the system's response to a step load transient from 75% to 25%. Again, a comparison of the waveforms of the two systems shows that the PWL LUT has no visible influence on the system's dynamic response.

With this positive simulation result in mind, we now have the confidence to implement the prototype.

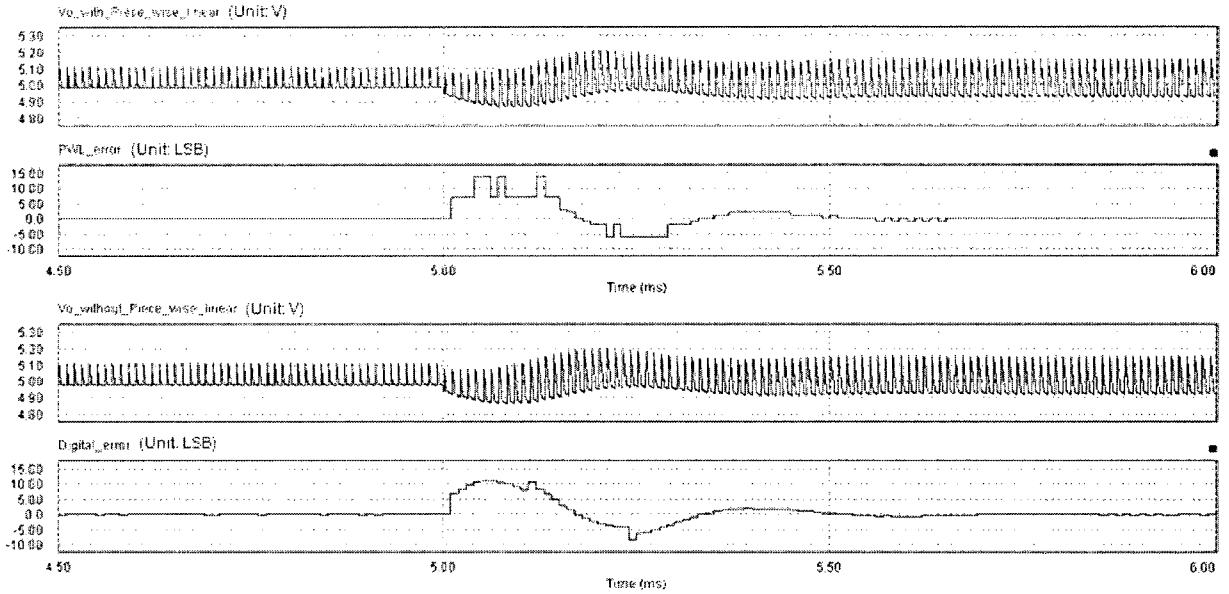


Figure 4-10 Load Transient Response 25% to 75% Load

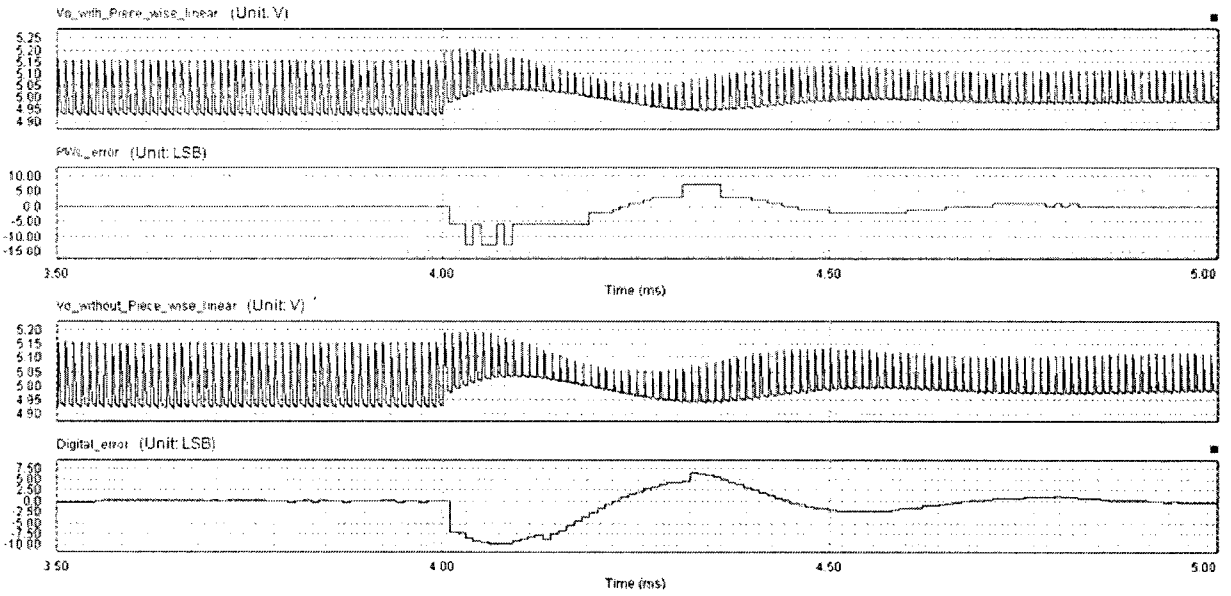


Figure 4-11 Load Transient Response 75% to 25% Load

Chapter 5 Implementation

5.1 Introduction

A prototype is built in order to validate the proposed system. The implemented system includes the proposed architecture and the proposed protocol that discussed in Chapter 3 and the Piece-wise Linear Lookup table and power stages that designed in Section 4.4. In this chapter, Section 5.2 is an overview of the implemented system. Section 5.3 describes the designed hardware. Section 5.4 describes the designed software. The PCB layout and the microcontroller source code are attached in Appendix A and Appendix B, respectively.

5.2 Implementation Overview

An overview of the implemented system is shown in Figure 5-1. The explanation is as follows.

There are two power stages on the board. They work at the same frequency.

Two 10-bit ADCs are used for sampling the output voltages at 4 times the PWM frequency. The first sample is synchronized at the beginning of each PWM cycle, and the following 3 samples are converted every 2 us.

The secondary-side controller calculates the average value of the 4 samples in each cycle in order to filter out the ripple voltage.

The averaged values of the two power stages are sent to a Piece-wise Linear Lookup table (LUT). The LUT converts 10-bit samples into 4-bit Piece-wise Linear codes.

The two 4-bit codes are combined into an 8-bit code and then encoded into a 10-bit code by an 8B/10B LUT. A serializer sends the 10-bit code to a transmitter and then through a pulse-transformer to a primary-side receiver.

If Over Voltage is detected by the secondary-side controller, a "Stop" code word will be immediately sent to the primary side. This has priority over all other codes.

A receiver on the primary side receives the 10-bit code. Ideally Clock Recovery method should be used, but it's not implemented in this system. A separate clock line is used instead.

A De-serializer deserializes the received bits into 10-bit bytes. A Word Aligner is used for recognizing special codes and frame synchronization. Once the Word Aligner detects a "Sync" code, within a byte or across two bytes, it reset the counter of the De-serializer, so that the data stream following the "Sync" code is synchronized.

The deserialized 10-bit byte is sent to the 10B/8B LUT. The 8-bit result is split into its original two 4-bit bytes.

The 4-bit bytes are translated back to 10-bit sampled values.

The primary-side controller calculates the next duty cycle based on the recently received sample values.

The primary- and secondary-side controllers are implemented by Microchip dsPIC30F2020 microcontrollers. The serializer, de-serializer, transmitter, receiver, and Word Aligner are implemented by separate digital chips.

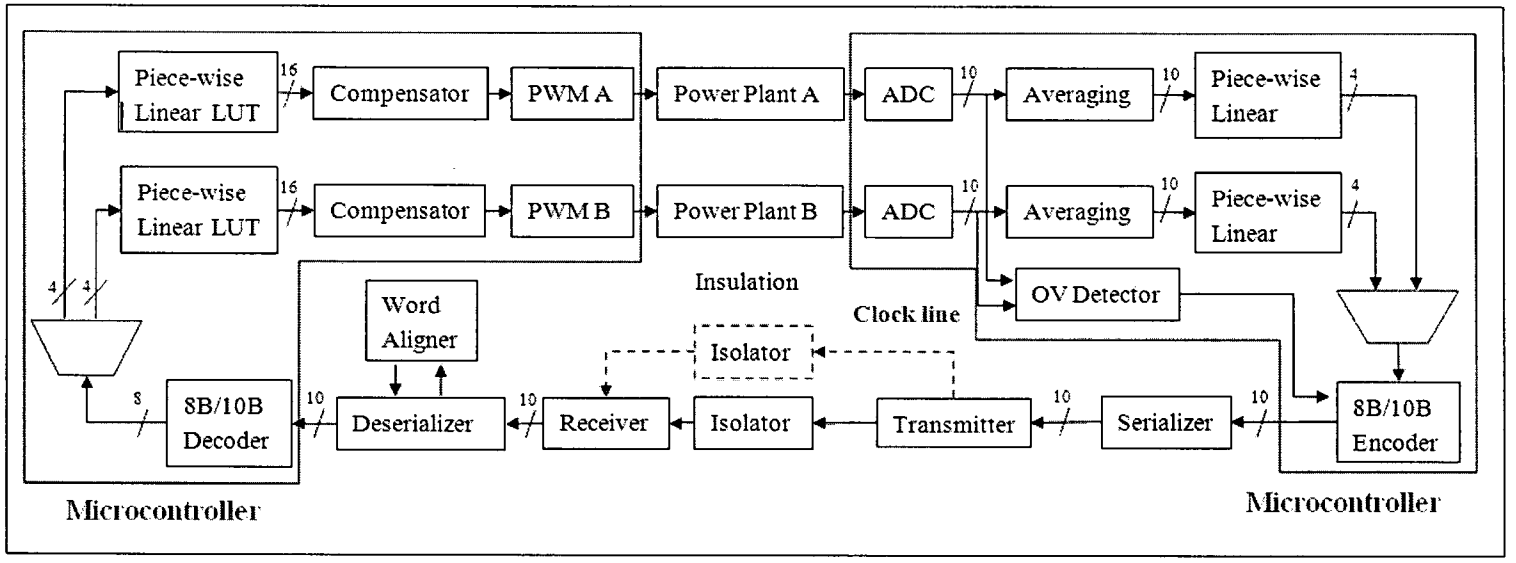


Figure 5-1 Implemented System Block Diagram

5.3 Hardware Design

The hardware design involves schematic design, PCB layout design, and component selecting. This chapter only discusses the schematic design. The PCB layout is attached in Appendix A. The Bill of Material is not included in this thesis.

5.3.1 Flyback Power Plants

The schematic of the designed power plants is shown in Figure 5-2.

The two Flyback power plants are shown in the upper and lower side of the schematic, respectively.

The input power enters the system from J1, through the common mode filters (L1 and L2), the transformers (T1 and T2), the output diodes (D9 and D10), and to the output connectors (J2 and J3).

The inductors L1 and L2 and the capacitors C93, C94, C82, C95, C96, and C83 comprise common mode filters, which can filter out possible voltage fluctuations caused by sudden load changes of the other power plant so that the two power plants will not interfere each other. L1 and L2's Pin 4 is referenced as primary-side ground.

The Zener diode VR3, transistor Q3, and resistors R83, R92, R93, R94, and R95 comprise an 11.6V voltage source supplying the MOSFET drivers, U27 and U28.

The MOSFET drivers, U27 and U28, receive PWM signals from the primary-side microcontroller, and drive the PWM pulses onto the Gates of the MOSFETs, Q1 and Q2.

R7, R8, R9, R10, R11, R12, R13, R14 are low-inductance current-sense resistors. When the current flowing through Q1 and Q2 exceeds a certain limit, a voltage droop on the

current-sense resistors will trigger the primary-side microcontroller to shut off for rest of the duty cycle. The threshold is programmable in the microcontroller.

C15, R5, R6, D7 and C17, R15, R16, D8 comprise a subber for each power plant. When the MOSFETs' turned off, the stored energy that resides in the leakage inductance will be discharged into the snubber and dissipated into heat.

The output voltages of power plants are Vout1 and Vout2. The voltage dividers (R77, R79 and R78, R80) are used to generate Over Voltage fault signals to the secondary-side microcontroller.

R81 and R82 comprise a voltage divider which generates a synchronization signal for the secondary-side microcontroller. The falling edge of this signal triggers 4 times of sampling in serial with 2uS interval. The diodes VZ1 and D11 prevent the SYNC signal from exceeding 5V or under 0V in order to protect the secondary-side microcontroller.

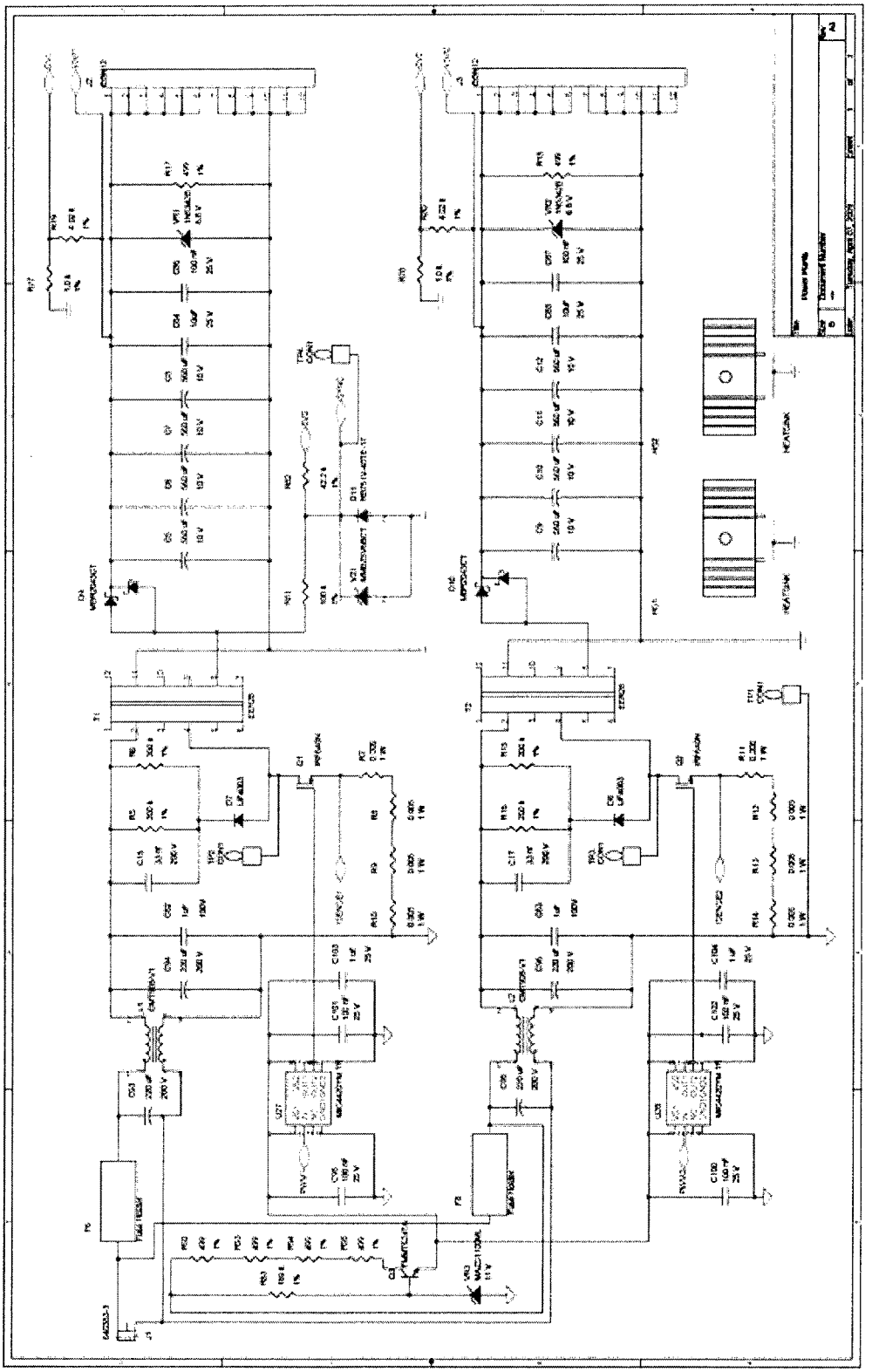


Figure 5-2 Power Plants Schematic

Because of the leakage inductance and various parasitic components, the voltage on secondary winding will oscillate between -5V to 5V after the energy is fully discharged and before the MOSFET is turned on. The falling edges of this oscillation will false trigger the synchronization signal if the threshold is set to above -5V. For this reason, the transformer turns ratio of 3:1 is chosen resulting in a trigger threshold of -8.5V; even the lowest input voltage, 36V, can provide a -12V falling edge on the secondary winding to distinguish itself from the ±5V ringing.

The Absolution Maximum Rating of the microcontroller pins is -0.3V to 5.3V. Although the divide ratio of the synchronization circuit is carefully designed, it cannot keep voltage in the safe range when an error happens. For example, when the communication between the two sides is lost and the output voltage goes up to 10V, there will be 6.48V on the microcontroller pin. In order to protect the microcontroller, the resistor values of the synchronization circuit must be very large in order to limit the current less than 20 mA in such cases.

The power plant design is the reuse of the design example in Section 4.4.

5.3.2 Secondary Control Circuit

The secondary-side control circuit is shown in Figure 5-3.

J4 is the Digital Signal Controller dsPIC30F2020.

J11 and J13 are SN74ABT841 (10-BIT BUS-INTERFACE D-TYPE LATCHES WITH 3-STATE OUTPUTS).

J12 is SN74ABT821 (10-BIT BUS-INTERFACE FLIP-FLOPS WITH 3-STATE OUTPUTS).

U25 is 74F160 (Synchronous Presettable BCD Decade Counter).

U13 is 74AC14 (Hex Inverter with Schmitt Trigger Input).

J11, J12, U25, and U13 comprise a buffer and a serializer for the output data from the microcontroller.

The microcontroller samples the output voltages from Pin2 and Pin3. It processes the data and put the resulting 10-bit code onto a parallel port. Its Pin 14 then sends out a pulse onto J11-Pin 13 to lock the data into J11. J11 stores the data, but the output port is held disabled until a pulse from U25, the decade counter, to enable it for 50 ns.

A 10MHz clock signal is generated from J4-Pin11, and is used globally in the control circuits. This clock signal ticks data out of the serializer, and is the input of U25 as well.

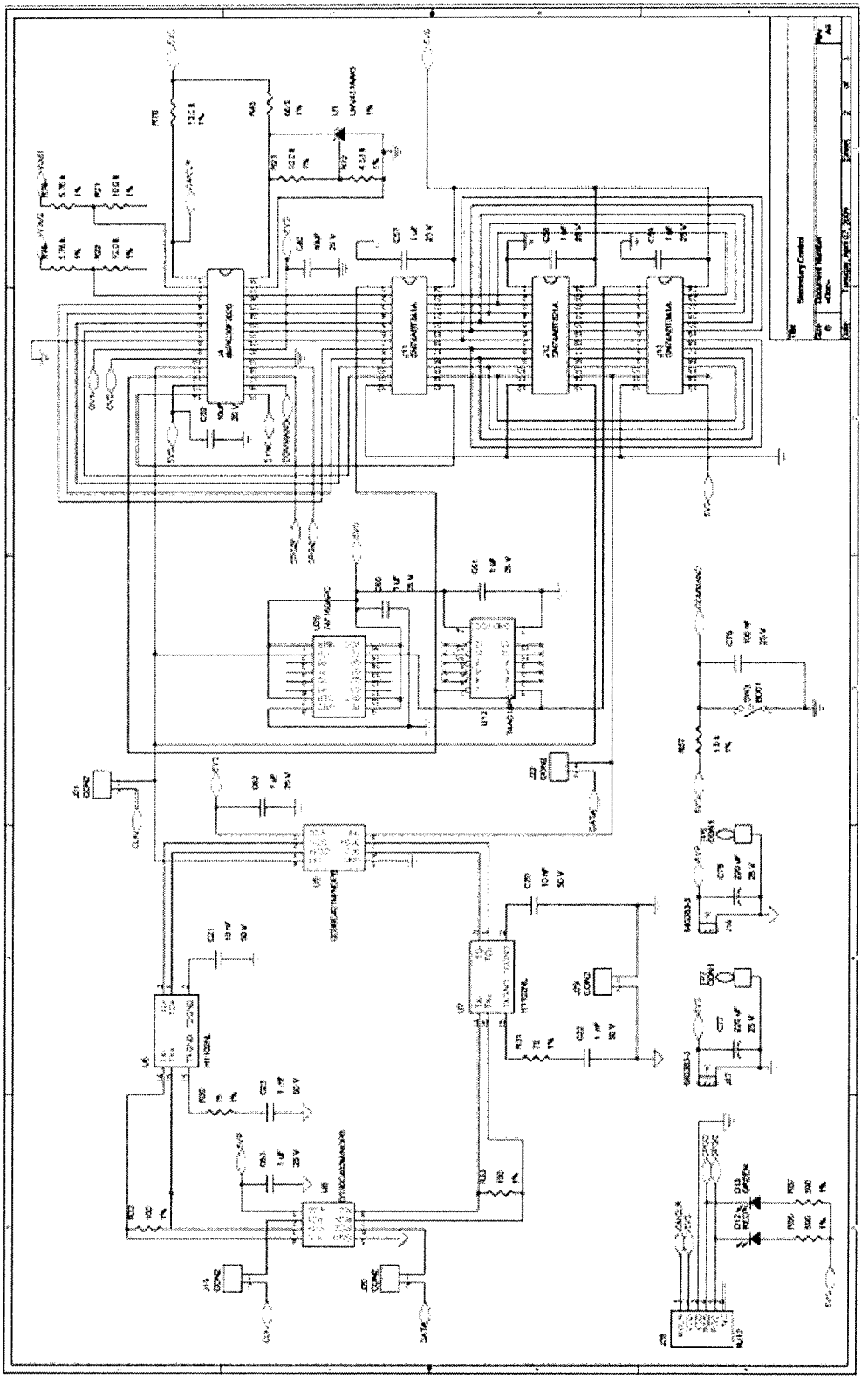


Figure 5-3 Secondary Control

Every 10 clock cycles, the decade counter, U25, sends out a pulse to enable J11's output and disable J13's output in order to shift a new 10-bit code from the buffer to the serializer.

U5 is DS90C401 (Dual Low Voltage Differential Signaling (LVDS) Driver).

U8 is DS90C402 (Dual Low Voltage Differential Signaling (LVDS) Receiver).

U6 and U7 are H1102NL (10/100BASE-T SINGLE PORT SURFACE MOUNT MAGNETICS).

U5, U6, U7, and U8 comprise an isolated data bus. U5 receives TTL signal and translates it into differential signal. U6 and U8 are pulse transformers. U6 carries clock signal; U7 carries data signal. U8 receives differential signal from U6 and U7, and translate into TTL signal to interface the primary-side control circuit.

The clock line can be removed if a phase-lock loop (PPL) is used on the primary side. In the implemented system, a separate clock line is used for simplicity purposes.

5.3.3 Primary Control Circuit

The secondary-side control circuit is shown in Figure 5-4.

J5 is the Digital Signal Controller dsPIC30F2020.

J15 is SN74ABT841 (10-BIT BUS-INTERFACE D-TYPE LATCHES WITH 3-STATE OUTPUTS).

J14 is SN74ABT821 (10-BIT BUS-INTERFACE FLIP-FLOPS WITH 3-STATE OUTPUTS).

U26 is 74F160 (Synchronous Presettable BCD Decade Counter).

J14, J15, and J26 comprise a deserializer. Every 10 clocks, the decade counter sends out a pulse to release the newly arrived 10-bit code from J15 to the microcontroller input ports. The microcontroller processes the data, and calculates the duty for the next cycle.

U11, U12, U17, U22, U23 are N74F133N (13-input NAND gate).

U15 and U16 are 74AC14 (Hex Inverter with Schmitt Trigger Input).

U15 and U16 invert the incoming data. U11, U12 together with U15 and U16 are used to recognize the "Sync" code from the data stream for frame synchronization. U22, U23 together with U15 and U16 are used to recognize the "stop" code in order to shut down PWMs in emergency.

U14 and U24 are 74F86SC (2-Input Exclusive-OR Gate). Each 8-bit code has two complementary forms in its 10-bit counterpart, and each NAND gate recognizes one of the two forms, therefore a XOR gate is connected after each pair of NAND gates.

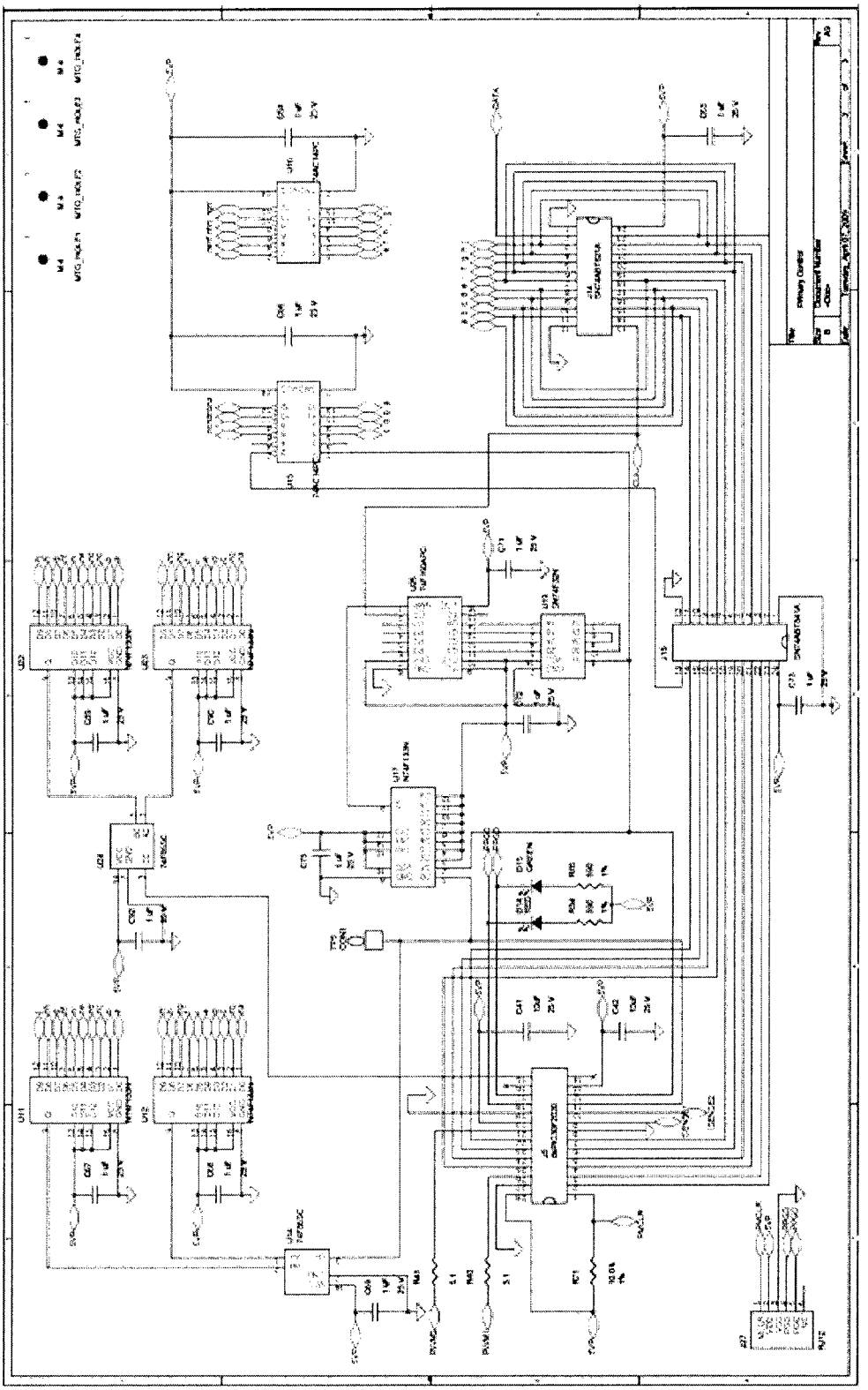


Figure 5-4 Primary Control

5.4 Software Design

5.4.1 Microcontroller Pin Maps

Microcontroller Pin Map (Secondary Side)

Pin Number	Pin Name	Function	Use
1	-MCLR	Master Clear	Connect to external +5V
2	AN0	Analog input channel①	Sample Channel 1
3	AN1	Analog input channel①	Sample Channel 2
4	RB2	Parallel Output	10B output bit5
5	RB3	Parallel Output	10B output bit6
6	RB4	Parallel Output	10B output bit7
7	RB5	Parallel Output	10B output bit8
8	V_{SS}	Ground reference for logic and I/O pins.	Connect to digital ground
9	CMP4A	Comparator 4 Channel A②	Over Voltage sense for Channel 1
10	CMP3D	Comparator 3 Channel D②	Over Voltage sense for Channel 2
11	PWM4H	PWM 4 High output	CP clock signal output
12	CN0	Input Change notification inputs ③	Receive "Request Data" signal
13	V_{DD}	Positive supply for logic and I/O pins	Connect to external +5V
14	RF6	Output	"Latch Memory" pulse output
15	IC1	Input Capture ④	Connect to SYNC Signal
16	RA9	Input ⑤	On/Off switch
17	RF7	Output	Drive LED 1
18	RF8	Output	Drive LED 2
19	V_{SS}	Ground reference for logic and I/O pins	Connect digital ground
20	V_{DD}	Positive supply for logic and I/O pins	Connect to external +5V
21	RE5	Parallel Output	10B output bit5
22	RE4	Parallel Output	10B output bit4
23	RE3	Parallel Output	10B output bit3
24	RE2	Parallel Output	10B output bit2
25	RE1	Parallel Output	10B output bit1
26	RE0	Parallel Output	10B output bit0
27	AV_{SS}	Ground reference for analog module	Connect to analog ground
28	AV_{dd}	Positive supply for analog module	Connect to LM431

Note:

① A 453 Ω +787 Ω resistive divider brings 5V output to 3.17V.

② Use the SMPS Comparator Module to detect OV fault at 1.18V through a resistive divider. The interrupt subroutine sends out the "Stop" code 0011111001 or 1100000110.

③ Enters interrupt when a negative pulse is received on Pin12. The subroutine outputs a new data on the parallel port, and sends out a positive pulse on Pin 14.

④ Triggered by falling edges on Pin 15. The subroutine turns on ADC sampling (4 times, 2 uS interval).

⑤ Logic Low is ON; Logic High is OFF.

Microcontroller Pin Map (Primary Side)

Pin Number	Pin Name	Function	Use
1	-MCLR	Master Clear	Connect to external +5V power supply
2	RB0	Parallel Input	10B input bit0
3	RB1	Parallel Input	10B input bit1
4	RB2	Parallel Input	10B input bit2
5	RB3	Parallel Input	10B input bit3
6	RB4	Parallel Input	10B input bit4
7	RB5	Parallel Input	10B input bit5
8	V_{SS}	Ground reference for logic and I/O pins	Connect digital ground
9	CMP3C	Comparator 3 Channel C①	Current limit for PWM1
10	CMP4B	Comparator 4 Channel B①	Current limit for PWM1
11	CN1	Input Change notification inputs ②	"Sync Code received" input
12			Reserved
13	V_{DD}	Positive supply for logic and I/O pins	Connect to external +5V power supply
14	INT2	External interrupt 2 ②	"Data Arrived" input
15	SFLT1	Shared Fault Pin 1 ③	"Stop Code Received" input
16			Reserved
17	RF7	Output ④	LED 1
18	RF8	Output ④	LED 2
19	V_{SS}	Ground reference for logic and I/O pins	Connect to digital ground
20	V_{DD}	Positive supply for logic and I/O pins	Connect to external +5V power supply
21	PWM3H	PWM 3 High output	PWM 2 output
22	RE4	Parallel Input	10B input bit9
23	RE3	Parallel Input	10B input bit8
24	RE2	Parallel Input	10B input bit7
25	RE1	Parallel Input	10B input bit6
26	PWM1L	PWM 1 Low output	PWM 1 output
27	AV_{SS}	Ground reference for analog module	Connect to analog ground
28	AV_{dd}	Positive supply for analog module	Connect to external +5V power supply

Note:

① Use SMPS Comparator Module, turn off the PWM for the current duty cycle when the input voltage exceeds 0.1V.

② "Data Arrived" interrupt and "Sync Code received" interrupt may happen at the same time. Responses to "Sync Code received" interrupt only.

③ "Data Arrived" interrupt and "Stop Code Received" interrupt may happen at the same time. Responses to "Stop Code Received" interrupt only.

④ Logic Low turns on the LED.

5.4.2 Program Description

1. Primary-side PWM

The primary-side microcontroller outputs 2 synchronized PWM signal at 100 KHz. The duties are independent.

2. Secondary-side ADC sampling

The Pin 15 of the secondary-side microcontroller receives the synchronization signal and triggers ADC sampling for 4 times in each PWM cycle. The time interval between two samples is 2 uS. The 2 channels are sampled simultaneously.

3. Averaging ADC samples

After capturing the 4^{th} sample, an average value is calculated for each channel in each cycle. (Right-shift 2 bits)

4. Piece-wise Linear LUT

Configure a 1024-byte LUT in the secondary-side microcontroller memory. Using the 10-bit average values as offset address, the corresponding 4-bit piece-wise linear code can be attained. The 4-bit code should be positioned in the 4 LSBs in each byte. The 4MSBs should be filled by 0s.

5. Assembling 8-bit Code

Combine the two 4-bit codes together to get an 8-bit code for 8B/10B encoding. The Channel 1 is positioned in 4 MSBs. The Channel 2 is positioned in 4 LSBs.

6. 8B/10B Encoding

Each 8B code has two corresponding 10B codes: one is in the RD+ group; the other is in the RD- group. Each group is stored in a 256-word (a word is 16 bit) LUT.

In each word, bit 0 ~ bit 9 stores the 10B code, and bit 15 stores the code's disparity. 0 in bit 15 stands for disparity neutral; 1 in bit 15 stands for disparity non-neutral. Codes in RD+ group feature -2 disparity or neutral disparity; codes in RD- group feature +2 disparity or neutral disparity. A complete 8B/10B table is in [44].

When encoding, the 8B code is used as offset address of the LUT. The choosing between RD+ LUT and RD- LUT is determined by the system's RD status: If the system's RD status is 1, then use the RD+ LUT; if the system's RD status is 0, then use the RD- LUT.

After sending out each 10B code, the program should update the system's RD status according to the disparity bit of the code that was just sent. If the disparity bit is 0, the system's RD status is left unchanged. If the disparity bit is 1, the system's RD status should be changed according to the table below:

Table 5-1 Disparity Update Table

Current Status	RD- Disparity 0	RD- Disparity 1	RD+ Disparity 0	RD+ Disparity1
0	Keep 0	Change to 1	———	———
1	———	———	Keep 1	Change to 0

7. Sending 10B Codes

The Pin 11 of the secondary-side microcontroller output an 8 MHz clock signal, driving the external transceiver circuit. The external circuit sends a "Request Data" pulse on Pin 12 every 10 clocks; the secondary-side microcontroller must put the new encoded number on the parallel port and send out a "Latch Memory" pulse on Pin 14 before the next "Request Data" pulse arrives.

A 10B code takes 1.25 uS to be sent out. Therefore in each PWM cycle (10uS), eight 10B codes can be sent. Each 10B code carries 2 channels, therefore except the first "Sync" code at the beginning of each frame; the system's capacity is 14 channels. In this implementation, the system only has two channels, and their feedback data are carried in the first byte after the "Sync" code. The rest of the data are not used by the primary-side microcontroller in this implementation.

8. Receiving 10B Codes

Once the primary-side digital circuit receives a 10B code, a "Data Arrived" pulse will be sent on the Pin 14. If the 10B code is the "Sync" code, a "Sync Code Received" pulse will be sent on Pin 11. In this implementation, open the "Data Arrived" interrupt only after receiving a "Sync Code Received" pulse, and disable to Data Arrived" interrupt as soon as the next 10B code is received.

9. 10B/8B Decoding

Configure a 1024-byte space for 10B/8B LUT. Use the 10-bit code as the address offset and get the corresponding 8-bit code.

10. Reverting 4B code

Upon decoding the 10B code into 8B, split the 8B code into two 4B code.

11. Piece-wise Linear Decoding

Configure a 16-word (double byte) LUT for Piece-wise Linear decoding. Each 4B code is used as address offset to look-up a 16-bit corresponding value. This 16-bit signed integer is not the same 10-bit sample value that was encoded, but rather the error value – the 10-bit sample value minus the reference voltage value. The 16-bit number is used in PID control algorithm and is referenced as e(k) in the formula.

12. PID Control Algorithm

Use the following formula to calculate the next duty cycle. Each channel is calculated separately.

$$D(k+1) = 36 \times e(k) + I(k) + 5 \times D(k) \tag{5.1}$$

$$I(k+1) = I(k) + e(k) \tag{5.2}$$

where D(k) is the next duty cycle, which is the number updated to the Duty Cycle Register; e(k) is the newly received error value, which is a 16 bit signed integer; I(k) is the value in the integrator, whose range is ± 261814.

D(k+1) is the number used in the Duty Cycle Register for the next duty cycle. Its effective range is +16～+2400 (Maximum 25% duty cycle by power plant design). When it's

less than 16 (including negative value), it is set to 16; and when it's larger than 2400, it is set to 2400.

I(k+1) is the integrator value for calculating the duty cycle following the next duty cycle. It should be updated after computing the next duty cycle.

13. System States and Indicators

The system has 4 states and 2 indicators to represent each state. The states are stored in a 2-bit memory.

Table 5-2 System States Table

System State	Description	State bits	Red LED	Green LED
Stop	Manual Stop or Fault Stop	00	Off	Off
Standby	Initialized, waiting for start	01	On	Off
Soft Start	During Start Sequence	10	Off	On
Normal	Normal Operation mode	11	On	On

14. Soft Start

The primary- and secondary- side microcontrollers set the System State to Standby after initialization. The secondary-side microcontroller is waiting for a "Start" command from an external button.

Once a "Start" command is received, the secondary-side microcontroller enters the Soft Start and continuously sends out RD+ and RD- "Sync" code alternately (RD+1100000101 and RD- 0011111010).

Once the secondary-side microcontroller receives the synchronization signal from the synchronization circuit, it starts to sample the output voltages and the system state is changed to Normal.

After initialization, the primary-side microcontroller is waiting for the "Sync" code as a start command. Once the "Sync" code is received, the primary-side microcontroller enters the Soft Start. The duties of the two PWMs increase by 1 LSB in each cycle. Once a feedback 4-bit sample is greater than 0111 (close to 5V), the corresponding integrator is initialized to a value that is computed from the current duty. Then the system enters the Normal state.

15. Stop Sequence

When the secondary-side microcontroller detects Over Voltage fault or receives a manual "Stop" command, it continuously sends out the "Stop" code. It detects the SYNC signal to confirm that if the power stages are shut down. Once the SYNC signal is absent for an extend period of time, the secondary-side microcontroller stops sending the "Stop" code and returns to the Standby State.

When the primary-side microcontroller receives a "Stop" code, it shut down the PWMs and returns to the Standby State.

5.4.3 **Program Flowchart**

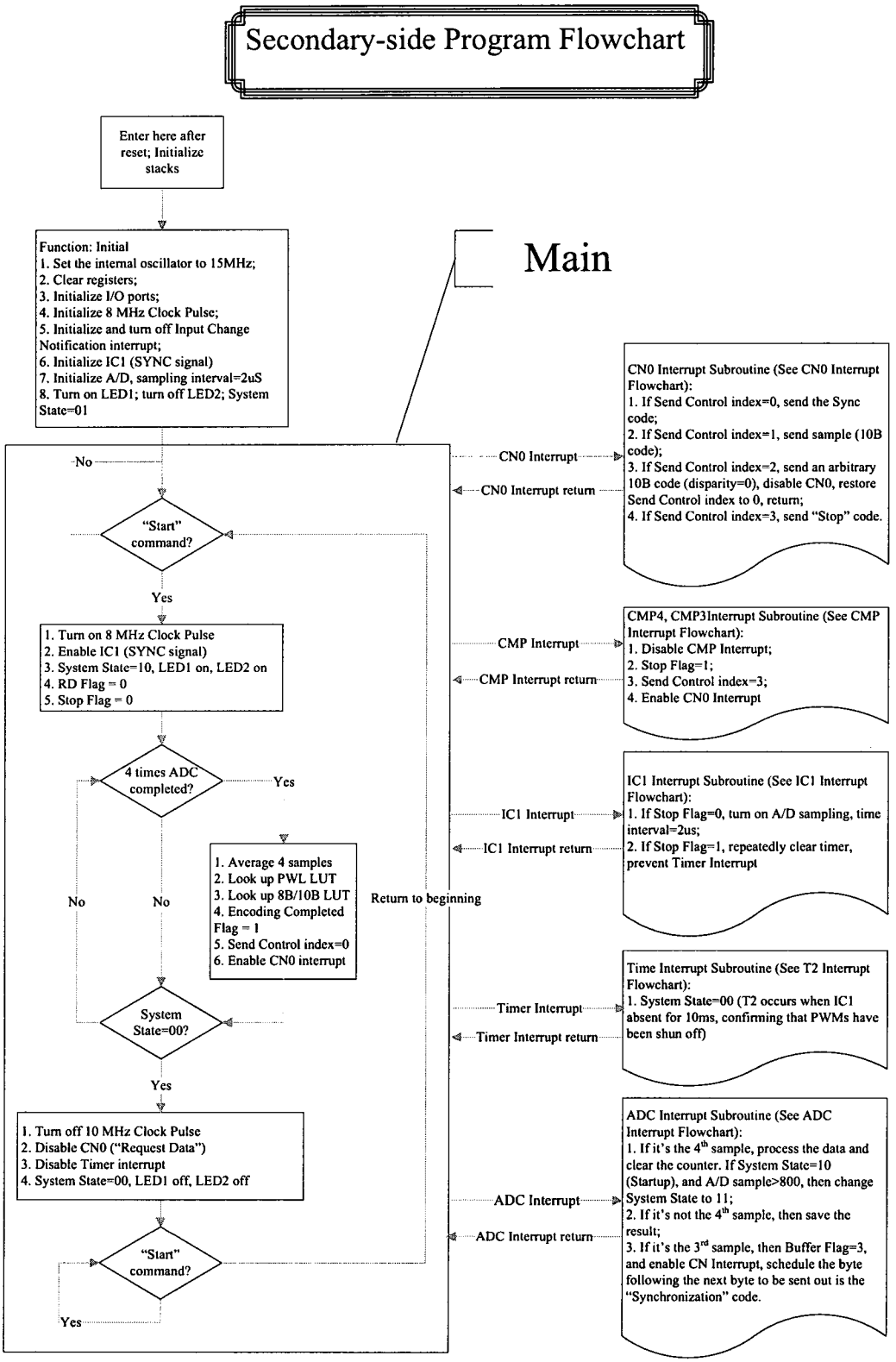


Figure 5-5 Secondary-side Program Flowchart

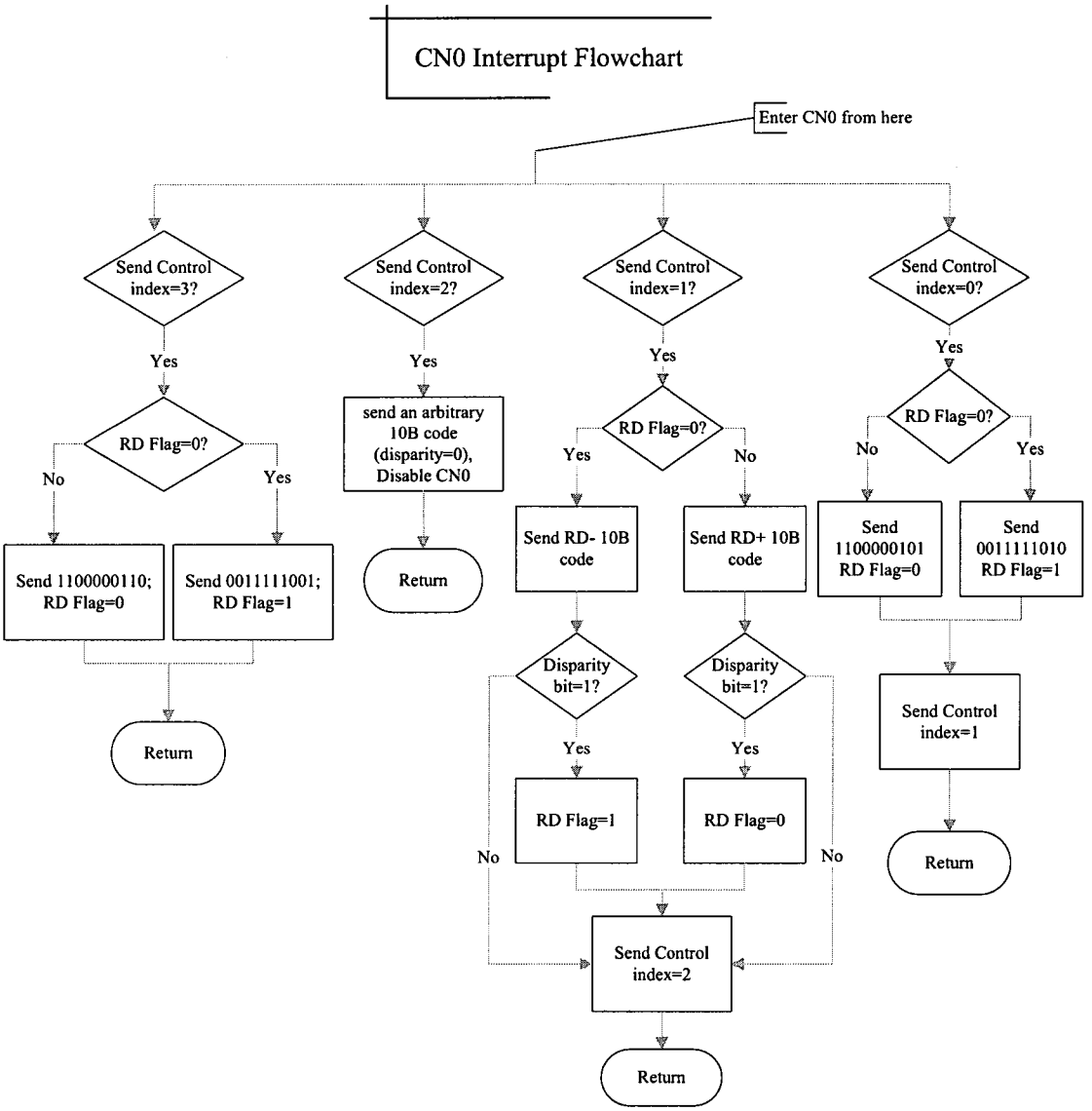


Figure 5-6 CN0 Interrupt Flowchart

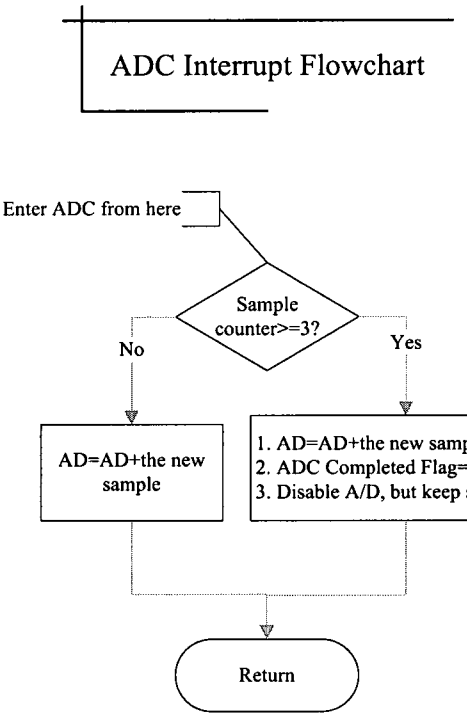


Figure 5-7 ADC Interrupt Flowchart

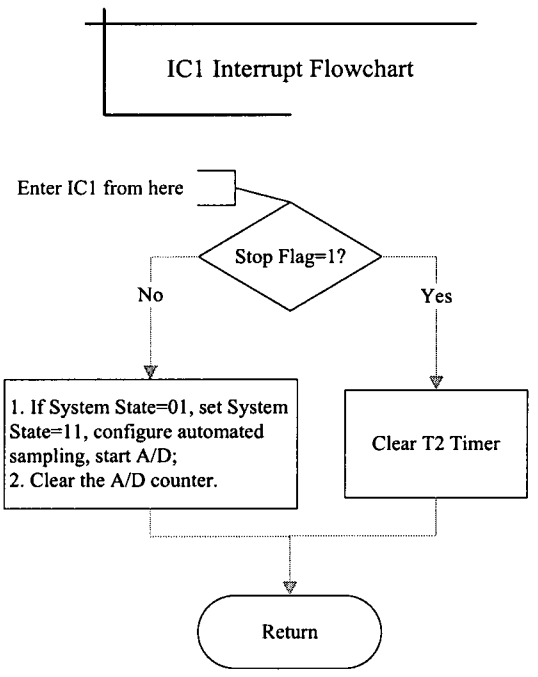


Figure 5-8 IC1 Interrupt Flowchart

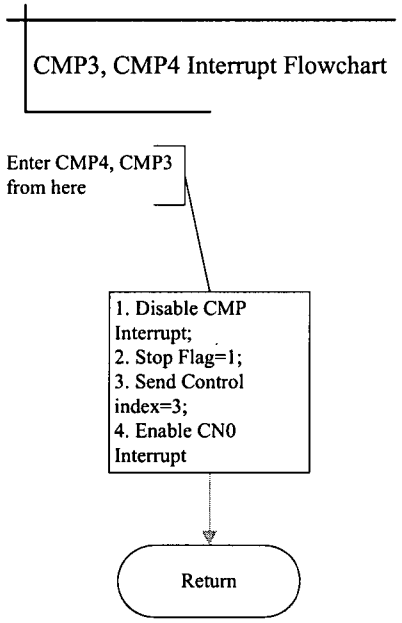


Figure 5-9 CMP3, CMP4 Interrupt Flowchart

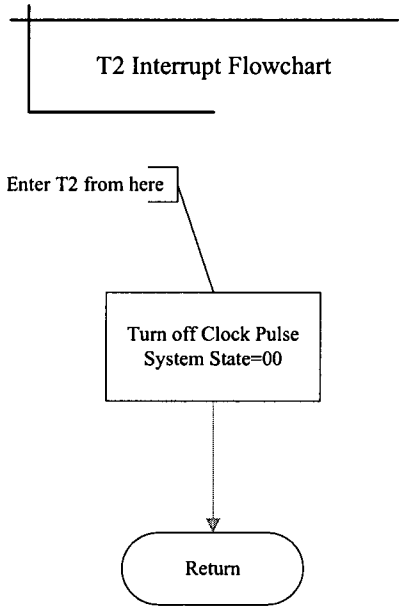


Figure 5-10 T2 Interrupt Flowchart

Primary-side Program Flowchart

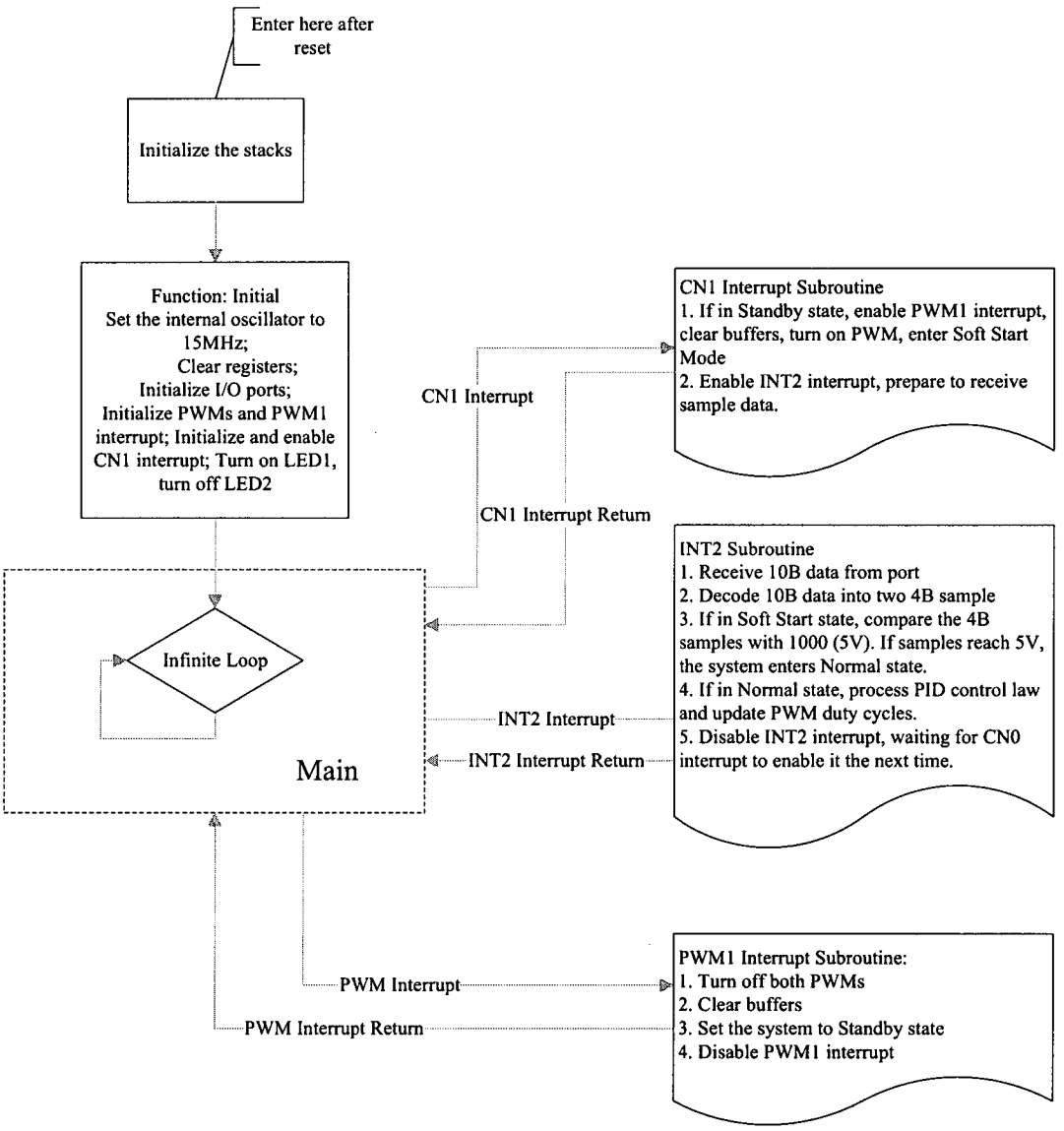


Figure 5-11 Primary-side Program Flowchart

Chapter 6 Experiment Results

6.1 Introduction

This chapter includes the test results of the implemented system. Section 6.2 shows the output performance. Output voltages of both power plants and the output current of Power Plant A are observed in various conditions to show the steady-state performance and the dynamic performance. Section 5 analyzes the data stream. The data line and the clock line waveforms are captured to extract the data stream. The "Sync Code received" signal is also monitored to help identify the beginning of each frame.

6.2 Output Performance

Test Condition

Constant resistive load is used in the test. Full load is 1 ohm, 75% load is 1.33ohm, 25% load is 4 ohm, and with 500 ohm on board.

In step load transient, the slope rate is set to 500mA/us.

Probe Setup

Channel 1: Vout1. Output voltage of the Power Plant A. Passive Voltage probe, DC coupled.

Channel 2: Vout2. Output voltage of the Power Plant B. Passive Voltage probe, DC coupled.

Channel 3: Vout1.AC_coupled. Output voltage of the Power Plant A. Differential probe, AC coupled.

Channel 4: Iout1. Output current of the Power Plant A. Current probe, DC coupled.

1. Steady State at full load

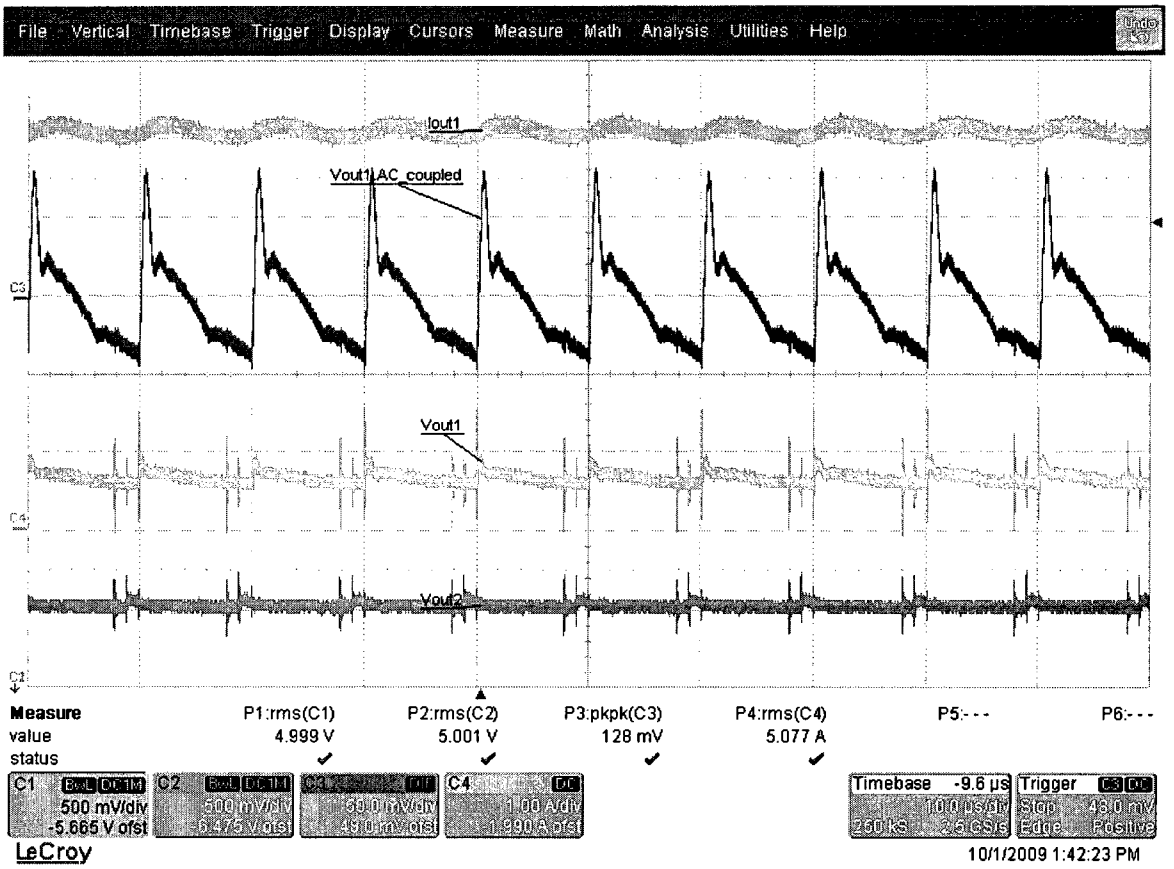


Figure 6-1 Steady State at Full Load

Vout1 and Vout2 are stabilized at 5V. The RMS value of Vout1 and Vout2 are 4.999V and 5.001V, respectively, which are very accurate. The ripple on Vout1 (peak to peak) is 128mV, which has nothing to do with neither the protocol nor the PID control, but the output capacitor and the power stage.

2. Soft Start at 25% Load

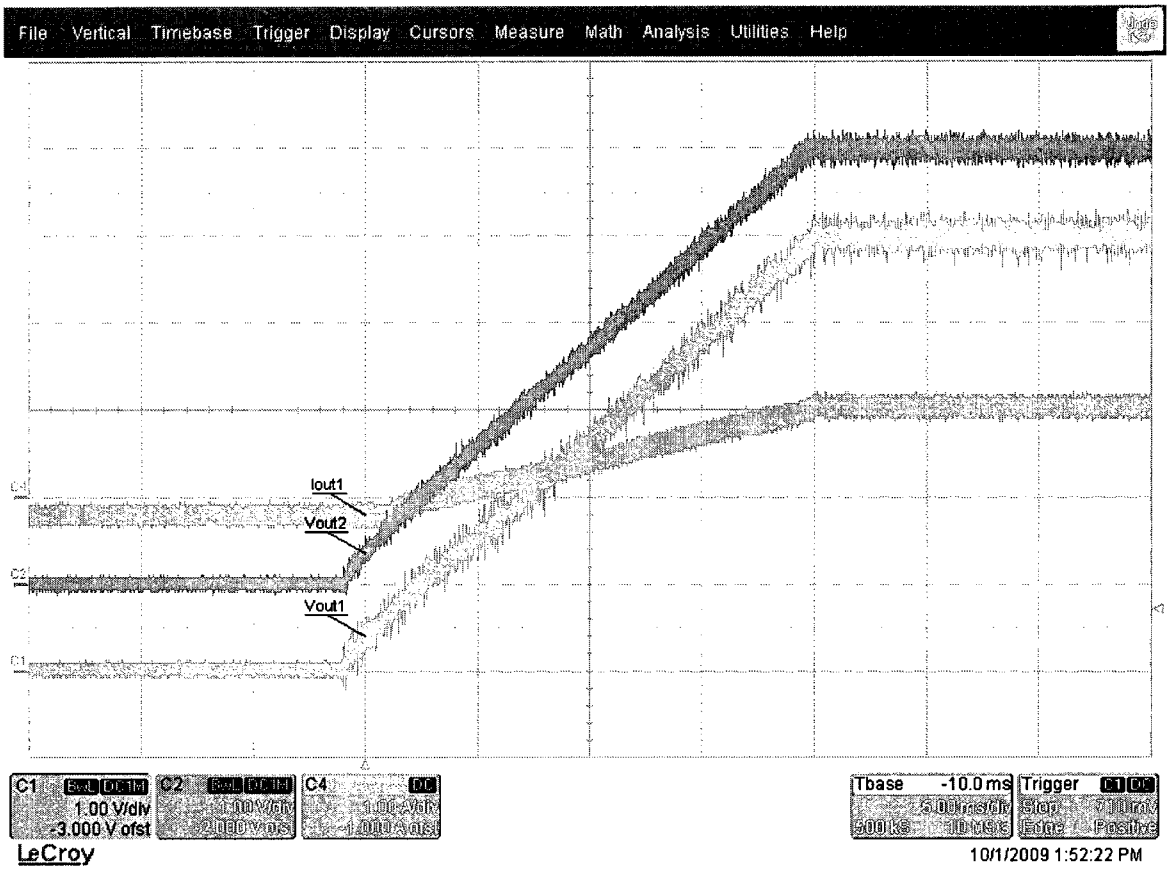


Figure 6-2 Soft Start

During soft start, the primary controller gradually increase the duty cycle until the output voltage reaches 5V, then the duty cycle value is divided by the integral coefficient Ki and the result is moved into the integrator of the PID controller. The PID controller then starts operating and the system quits the soft start state.

From the waveform one can see that the soft start takes around 20ms, which is a proper value. A too fast start-up is not desired, as it will excessively stress the output capacitor.

No overshoot is observed. The soft start is successful.

3. Load Transient Response 75% to 25% Load

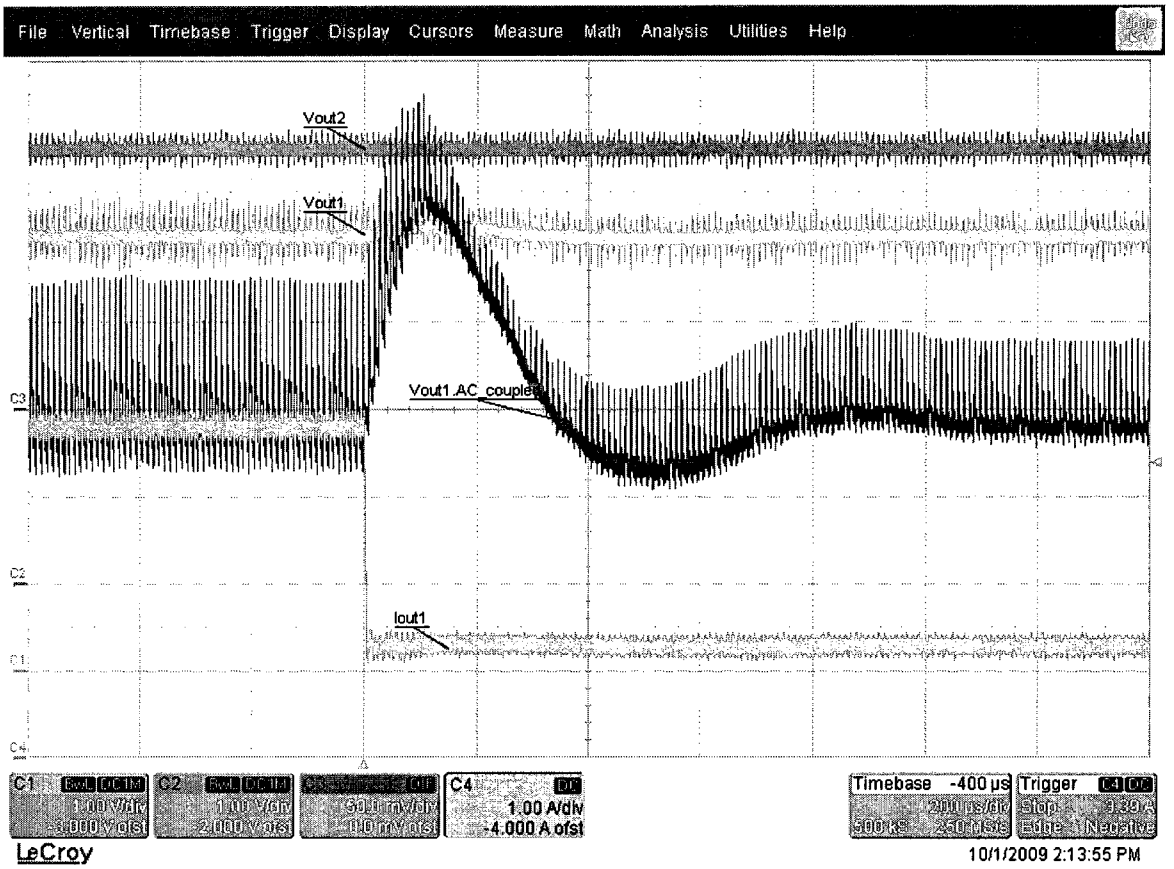


Figure 6-3 Load Transient Response 75% to 25% Load

At the step load transient, the output voltage overshoots to 5.1V, and then undershoots to above 4.95V, and returns to steady state. Although the response can be improved by tuning the PID controller, it is not the thesis' objective. This waveform shows that the error voltage during a load transient does not exceed the ±5% output dynamic range; it proves that the Piece-wise Linear Feedback method provides adequate error range for operation and the method is valid. This waveform also shows no visible nonlinear characteristic during the transient response; it proves that the Piece-wise Linear Method does not degrade the

system's dynamic response. Lastly, the Vout2 is stable during the load transient on Vout1; this proves that the multiplexed feedback solution is effective.

4. Load Transient Response 25% to 75% Load

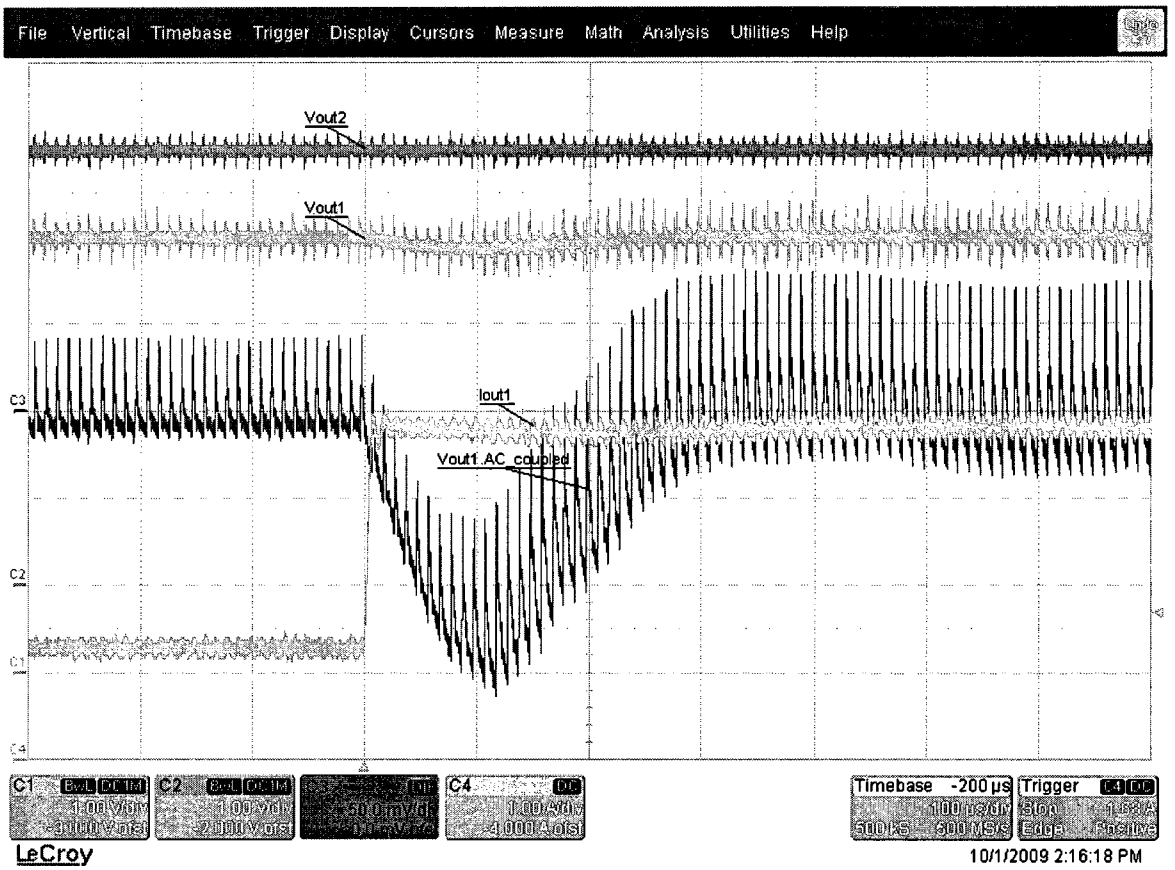


Figure 6-4 Load Transient Response 25% to 75% Load

During the step load transient, the output voltage drops to no less than 4.85V, and is quickly recovered within 300 us. Although the response can be improved by tuning the PID controller, it is not the thesis' objective. Similar to Figure 6-3, this waveform proves that the Piece-wise Linear Feedback method provides adequate error range for operation, the Piece-wise Linear Method does not degrade the system's dynamic response and the multiplexed feedback solution is effective.

5. Output Regulation Test

Figure 6-5 demonstrates the excellent output accuracy of the proposed system. Throughout the input voltage range and the output current range, the output voltage is regulated within 1 LSB from the rated 5V.

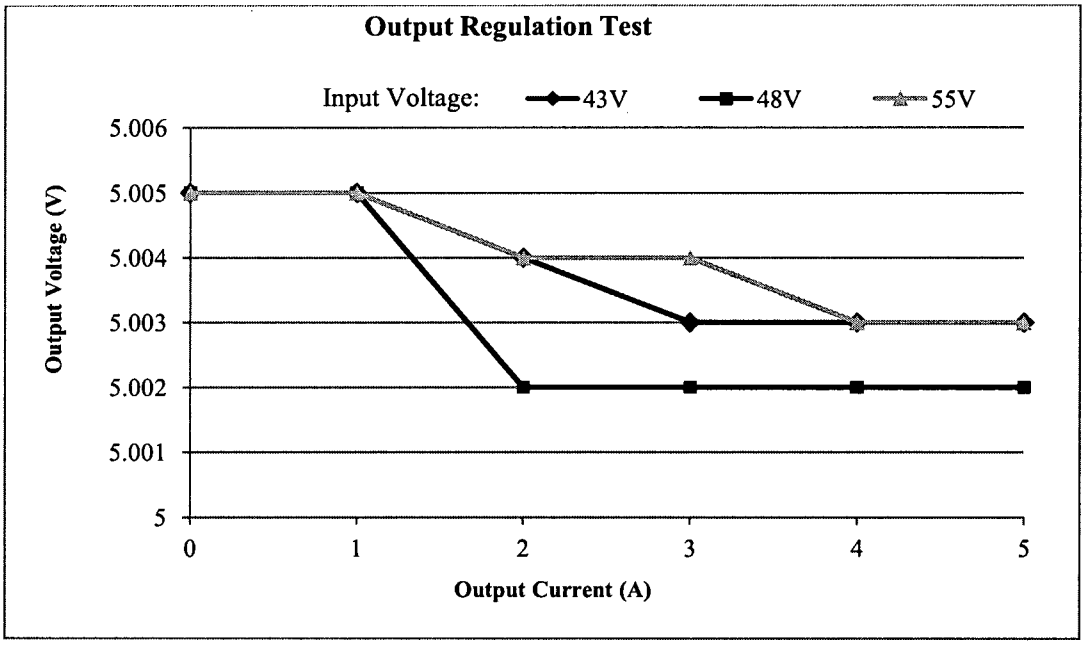


Figure 6-5 Output Regulation Test

6.3 Data Stream Analysis

Data stream in 9 successive frames are captured in a single waveform. The waveform is then zoomed in at each of the 9 frames in order to examine the exact data sent from the secondary side to the primary side. This study helps us to gain insight of the operating mode of the prototype board, and to verify the DC-balance as well.

Test Condition

Steady state at full load

Probe Setup

Channel 1: Data line. Passive Voltage probe, DC coupled.

Channel 2: Clock line. Passive Voltage probe, DC coupled.

Channel 3: "Sync Code received" pin. Passive Voltage probe, DC coupled.

The "Sync" code is for frame synchronization. It helps us to identify the beginning of each frame. The first 10B code after the "Sync" code carries the sample data of two power stages implemented in this system. The following six 10B codes in each frame are filled with "01100 01011" (corresponding to "0000 0000" in 8B code), which can carry samples for additional 12 power stages that are not implemented in this prototype system.

The upper half of each waveform is the snapshot of the data stream in 9 successive switching cycles, which is identical through Figure 6-6 to Figure 6-14. The lower half of each waveform is the zoom-in display of each of the 9 samples. The zoomed-in area is highlighted in the upper half of the waveforms.

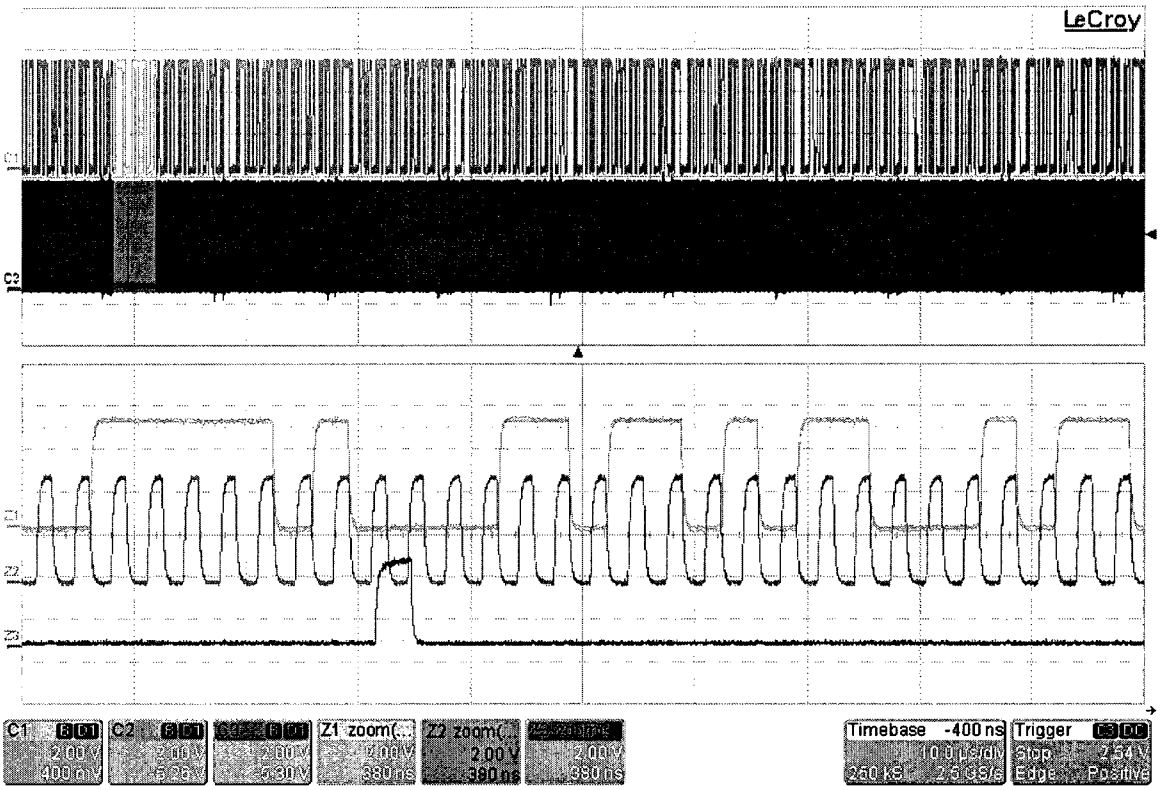


Figure 6-6 Data stream (zoom-in at the 1st sample)

Figure 6-6 contains three 10B codes: (from left) 00111 11010, 00011 01101 and 01100 01011.

Decoding the 10B codes we get:

Table 6-1 Decoding of the 1st sample

10B code	8B code	Disparity	Notes
00111 11010	Control code	+2	"Sync" code, beginning of the frame
00011 01101	1000 1000	0	Sample
01100 01011	0000 0000	0	Not used

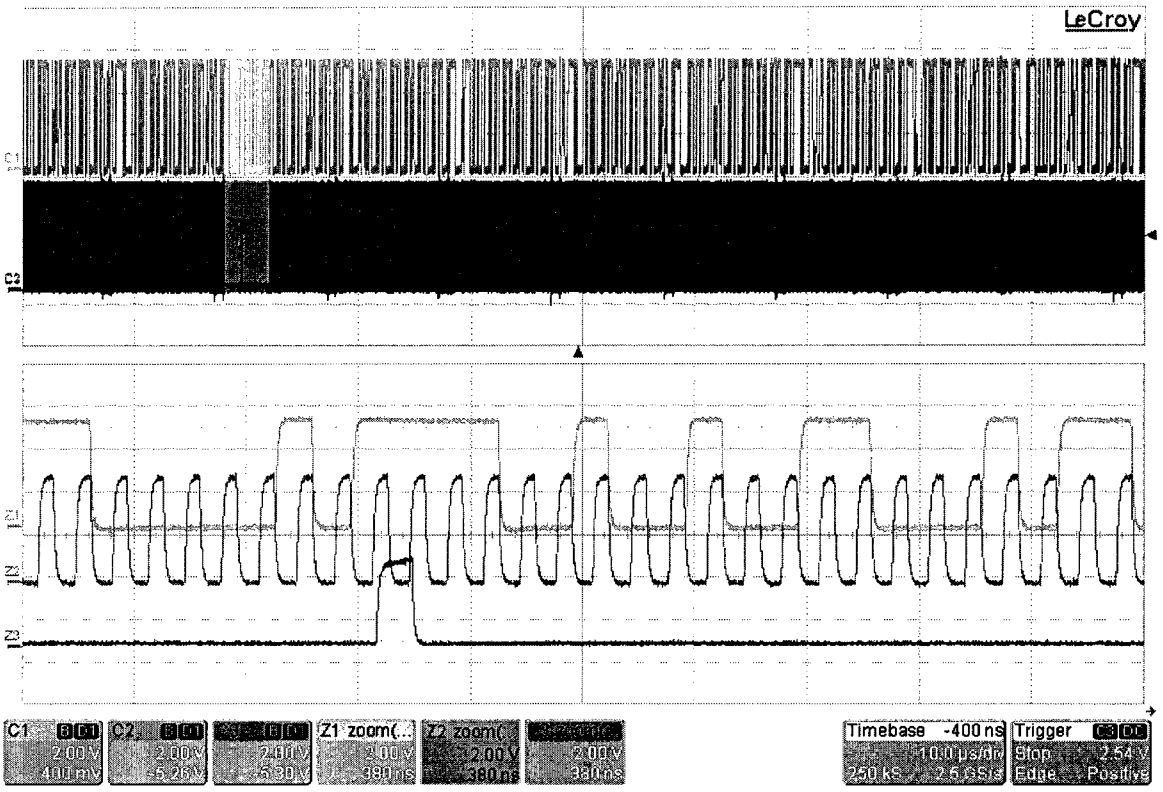


Figure 6-7 Data stream (zoom-in at the 2nd sample)

Figure 6-7 contains three 10B codes: (from left) 11000 00101, 11100 10010 and 01100 01011.

Decoding the 10B codes we get:

Table 6-2 Decoding of the 2nd sample

10B code	8B code	Disparity	Notes
11000 00101	Control code	-2	"Sync" code, beginning of the frame
11100 10010	1000 1000	0	Sample
01100 01011	0000 0000	0	Not used

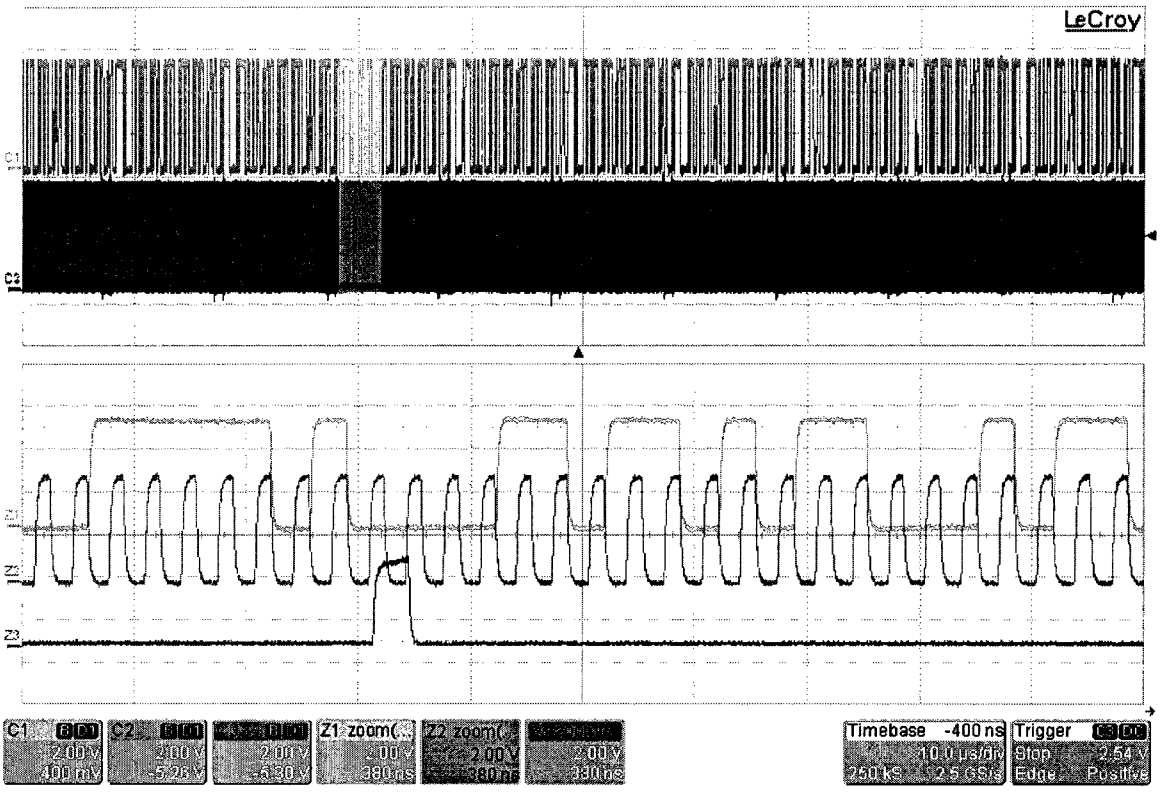


Figure 6-8 Data stream (zoom-in at the 3rd sample)

Figure 6-8 contains three 10B codes: (from left) 00111 11010, 00011 01101 and 01100 01011.

Decoding the 10B codes we get:

Table 6-3 Decoding of the 3rd sample

10B code	8B code	Disparity	Notes
00111 11010	Control code	+2	"Sync" code, beginning of the frame
00011 01101	1000 1000	0	Sample
01100 01011	0000 0000	0	Not used

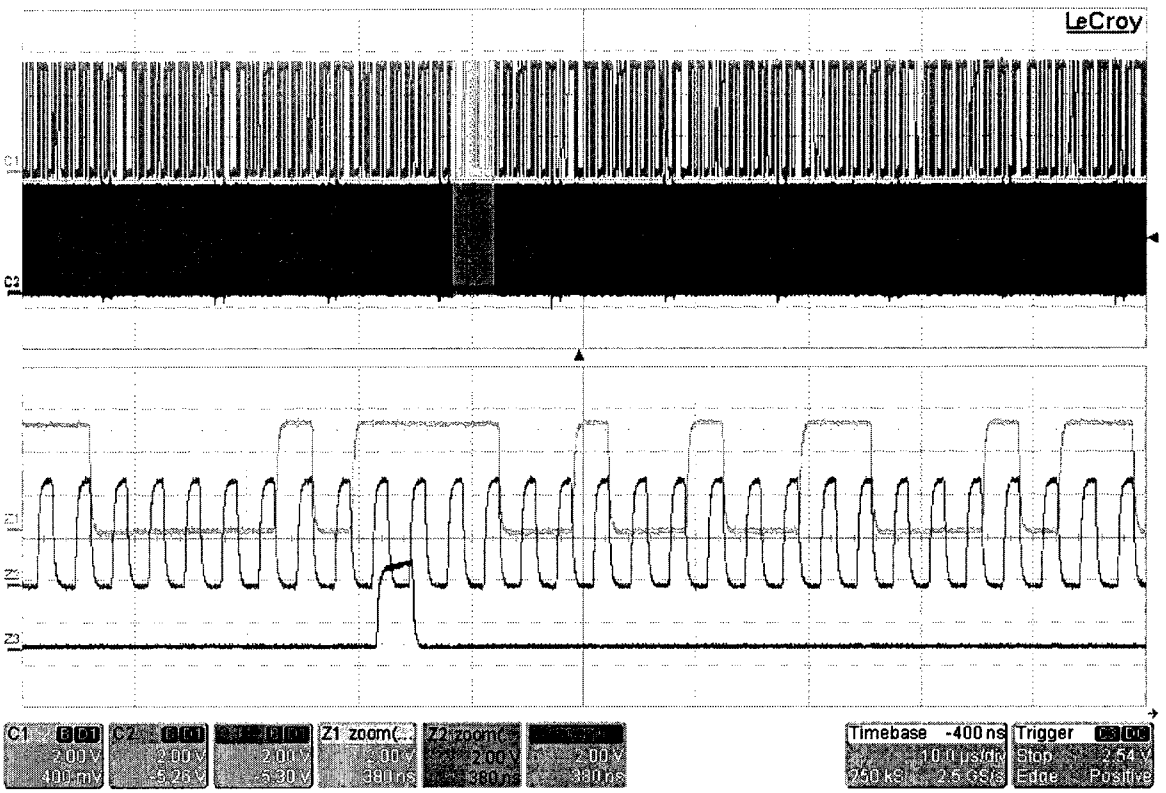


Figure 6-9 Data stream (zoom-in at the 4th sample)

Figure 6-9 contains three 10B codes: (from left) 11000 00101, 11100 10010 and 01100 01011.

Decoding the 10B codes we get:

Table 6-4 Decoding of the 4th sample

10B code	8B code	Disparity	Notes
11000 00101	Control code	-2	"Sync" code, beginning of the frame
11100 10010	1000 1000	0	Sample
01100 01011	0000 0000	0	Not used

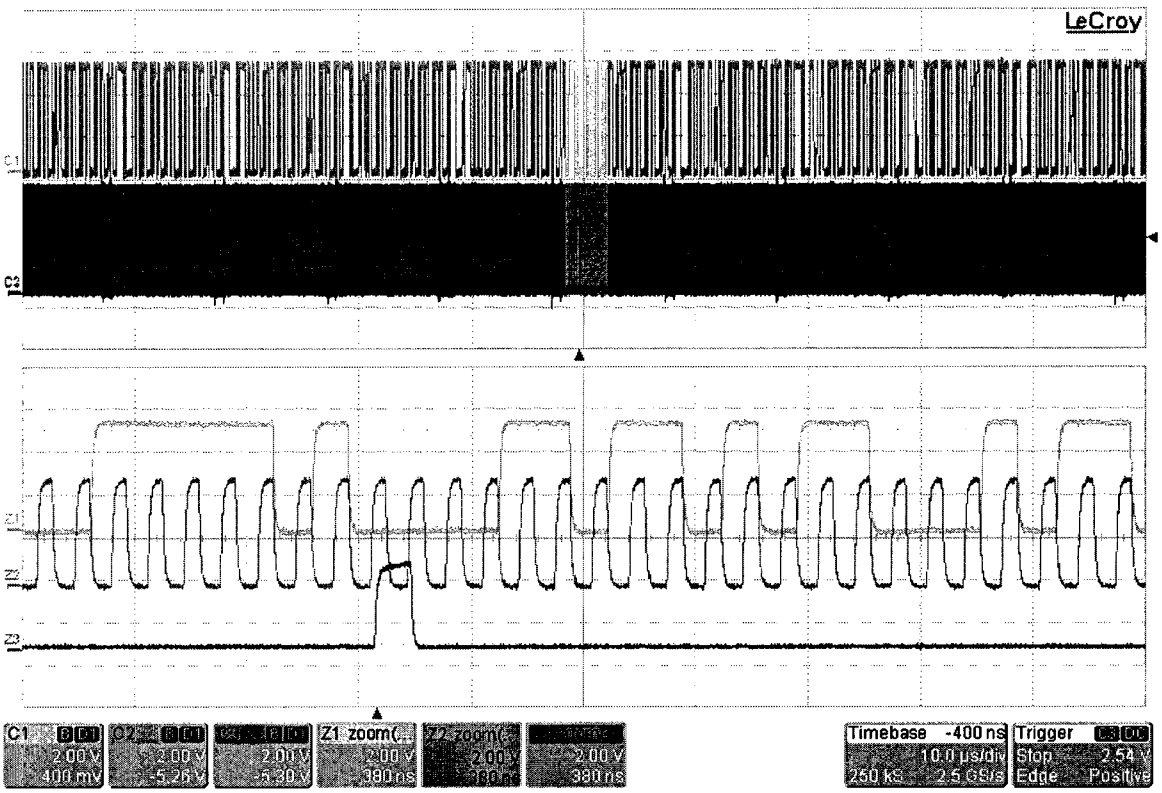


Figure 6-10 Data stream (zoom-in at the 5th sample)

Figure 6-10 contains three 10B codes: (from left) 00111 11010, 00011 01101 and 01100 01011.

Decoding the 10B codes we get:

Table 6-5 Decoding of the 5th sample

10B code	8B code	Disparity	Notes
00111 11010	Control code	+2	"Sync" code, beginning of the frame
00011 01101	1000 1000	0	Sample
01100 01011	0000 0000	0	Not used

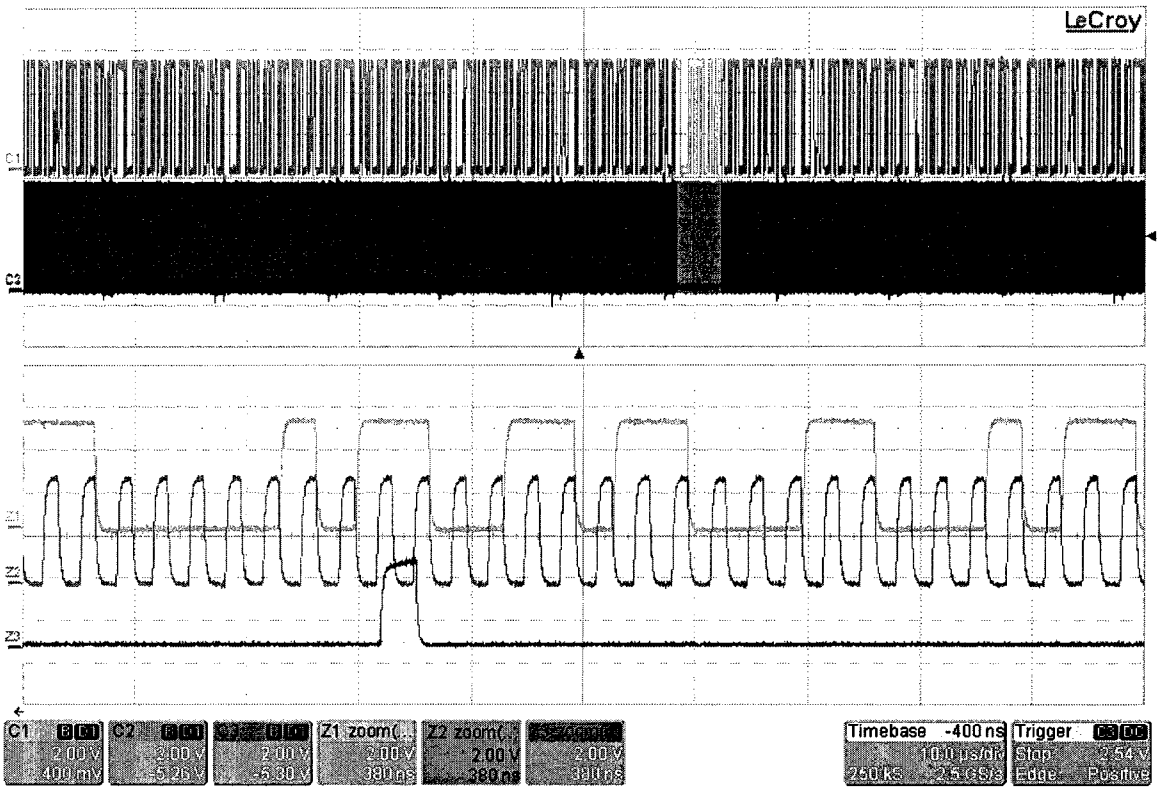


Figure 6-11 Data stream (zoom-in at the 6th sample)

Figure 6-11 contains three 10B codes: (from left) 11000 00101, 10011 01100 and 01100 01011.

Decoding the 10B codes we get:

Table 6-6 Decoding of the 6th sample

10B code	8B code	Disparity	Notes
11000 00101	Control code	-2	"Sync" code, beginning of the frame
10011 01100	0111 1001	0	Sample
01100 01011	0000 0000	0	Not used

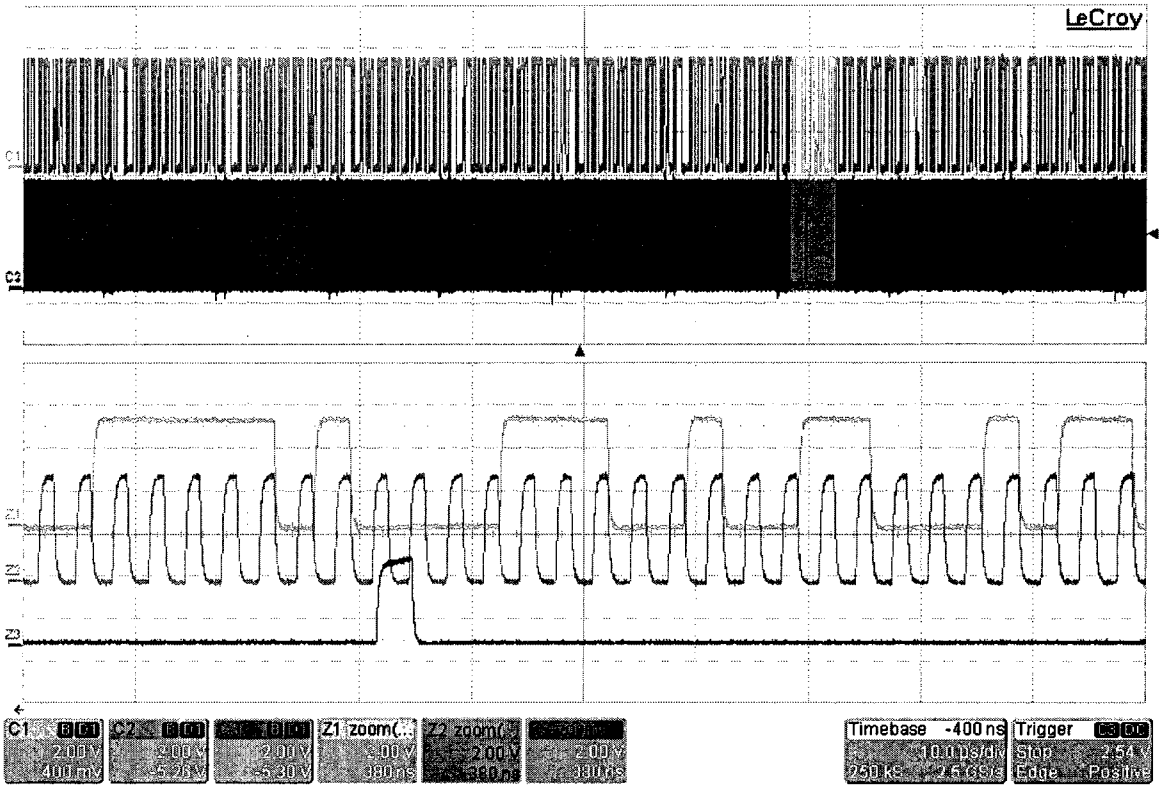


Figure 6-12 Data stream (zoom-in at the 7th sample)

Figure 6-12 contains three 10B codes: (from left) 00111 11010, 00011 10010 and 01100 01011.

Decoding the 10B codes we get:

Table 6-7 Decoding of the 7th sample

10B code	8B code	Disparity	Notes
00111 11010	Control code	+2	"Sync" code, beginning of the frame
00011 10010	1000 0111	-2	Sample
01100 01011	0000 0000	0	Not used

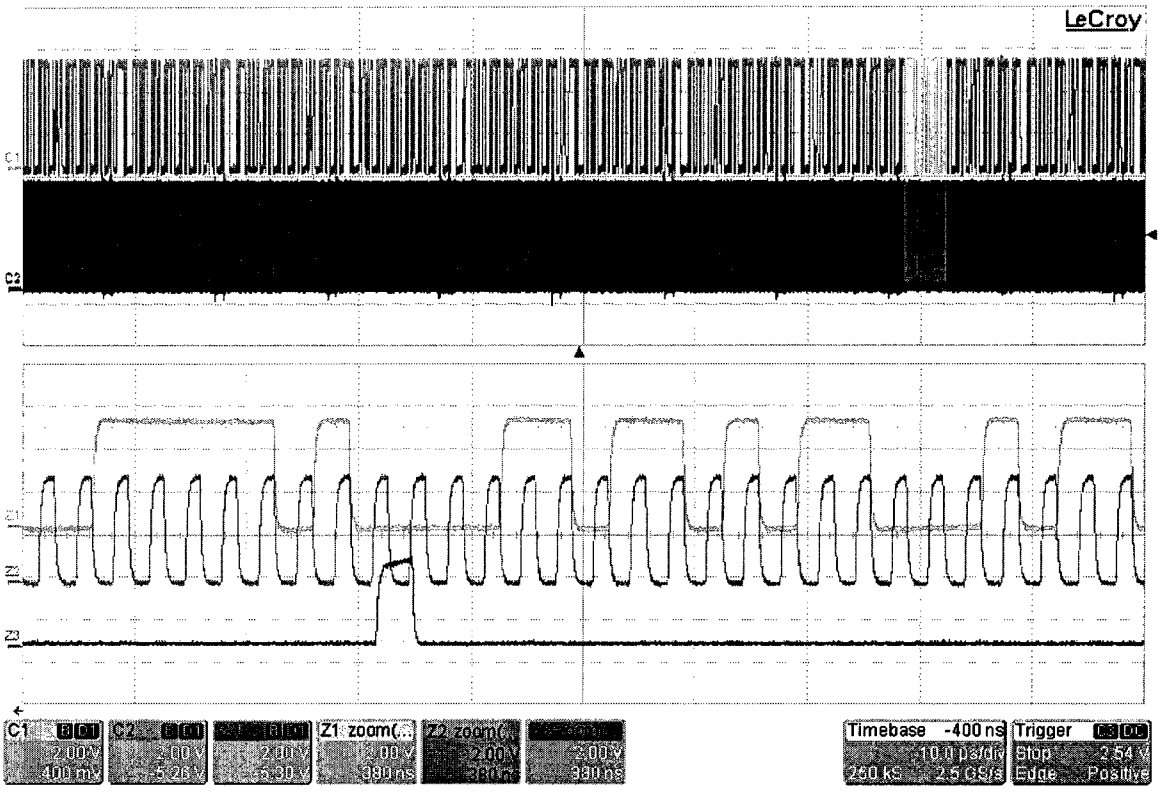


Figure 6-13 Data stream (zoom-in at the 8th sample)

Figure 6-13Figure 6-10 contains three 10B codes: (from left) 00111 11010, 00011 01101 and 01100 01011.

Decoding the 10B codes we get:

Table 6-8 Decoding of the 8th sample

10B code	8B code	Disparity	Notes
00111 11010	Control code	+2	"Sync" code, beginning of the frame
00011 01101	1000 1000	0	Sample
01100 01011	0000 0000	0	Not used

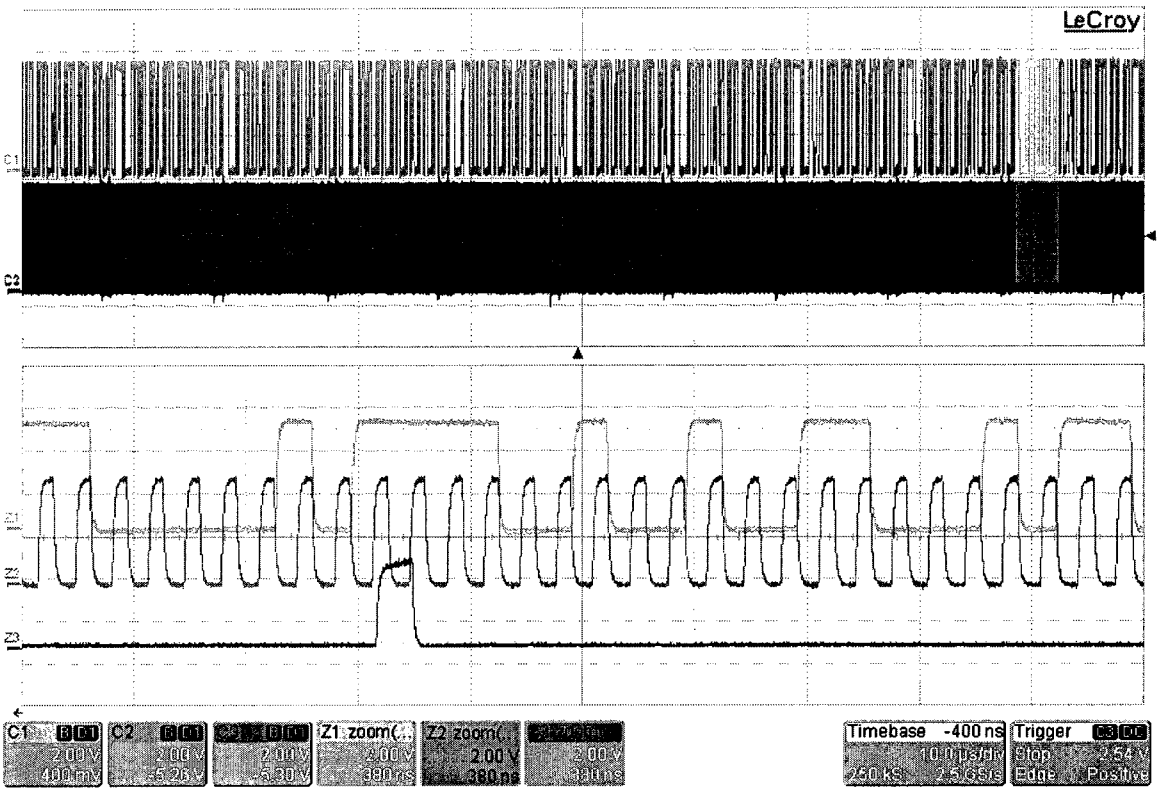


Figure 6-14 Data stream (zoom-in at the 9th sample)

Figure 6-14 contains three 10B codes: (from left) 11000 00101, 11100 10010 and 01100 01011.

Decoding the 10B codes we get:

Table 6-9 Decoding of the 9th sample

10B code	8B code	Disparity	Notes
11000 00101	Control code	-2	"Sync" code, beginning of the frame
11100 10010	1000 1000	0	Sample
01100 01011	0000 0000	0	Not used

The 9 successive samples are summarized in Table 6-10.

Table 6-10 Nine Successive PWL Samples

Sample	Power Plant A	Power Plant B
1	1000	1000
2	1000	1000
3	1000	1000
4	1000	1000
5	1000	1000
6	0111	1001
7	1000	0111
8	1000	1000
9	1000	1000

As it is shown in Table 4-1, the corresponding 4-bit code of reference voltage is 1000. Therefore Table 6-10 shows that the output voltages of both Power Plant A and Power Plant B are controlled at the reference voltage at most of the time, and occasionally vary within ±1 LSB.

Ideally, the output voltages should always be at the reference voltage in steady state. The ±1 LSB variation may be caused by noise or the ripple. Nevertheless, this result is still very accurate and of high quality comparing to other isolated digital power supplies. The thesis objective is met.

Waveforms from Figure 6-6 to Figure 6-14 also show a good example of DC-balance. The disparity of the "Sync" code of the 1st frame is +2, and the disparities of the rest 10B codes in that frame are 0. Therefore the disparity of the "Sync" code of the 2nd frame is chosen to be -2. And for the same reason, the disparities of the "Sync" code of the 3rd, 4th, 5th, 6th, 7th frames are +2, -2, +2, -2, +2, respectively. The disparity of the sample code of the 7th

frame is chosen to -2 because the 7th "Sync" code's disparity is +2. The rest codes in the 7th frame are disparity 0, therefore the disparities of the "Sync" code of the 8th frame is +2, and similarly, the disparities of the "Sync" code of the 9th frame is -2.

As an additional proof to show the codes following the 2nd 10B code in each frame are filled with 01100 01011 (corresponding to 0000 0000 in 8B code), Figure 6-15 shows the entire data stream of the 7th frame.

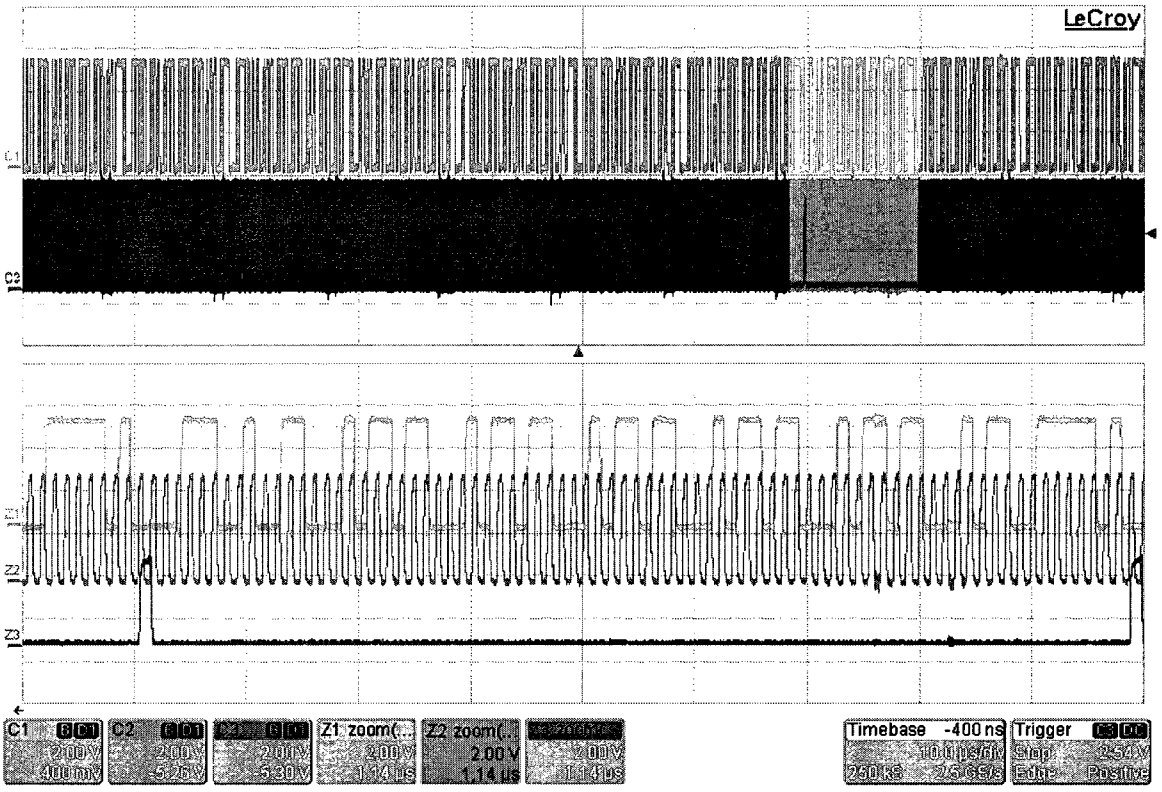


Figure 6-15 Data steam (zoom-in of the entire 7th frame)

Figure 6-15 contains the entire 7th frame and the "Sync" code of the 8th frame: (from left) 00111 11010, 00011 10010, 01100 01011, 01100 01011, 01100 01011, 01100 01011, 01100 01011, 01100 01011, 00111 11010.

Decoding the 10B codes we get:

Table 6-11 Decoding of the entire 7th frame

10B code	8B code	Disparity	Notes
00111 11010	Control code	+2	"Sync" code, beginning of the frame
00011 10010	1000 0111	-2	Sample
01100 01011	0000 0000	0	Not used
01100 01011	0000 0000	0	Not used
01100 01011	0000 0000	0	Not used
01100 01011	0000 0000	0	Not used
01100 01011	0000 0000	0	Not used
01100 01011	0000 0000	0	Not used
00111 11010	Control code	+2	"Sync" code, beginning of the next frame

The test results show that, the disparities of the 9 successive frames (72 successive 10B codes) strictly keep DC-balance.

Chapter 7 Conclusion and Future Work

7.1 Conclusion

This thesis proposed a multiplexed feedback solution for isolated digital power supplies, which is optimized for reduced cost, improved accuracy, and enhanced control functions. The proposed solution includes three parts: the architecture, the protocol, and the Piece-wise Linear Feedback method.

Chapter 1 stated the motivations and objectives of this thesis, which are to provide cost and performance benefits for isolated digital power supply systems. Chapter 2 reviewed the background knowledge in digital power supplies, and discussed the architectures and limitations of existing designs. Chapter 3 proposed a new system, as well as the considerations in its protocol design. Chapter 4 theoretically analyzed the Piece-wise Linear Feedback method and gave a design example. Computer simulation results support the theoretical analysis. Chapter 5 described the implementation of the proposed system. Chapter 6 demonstrated the satisfactory experiment performance.

The overall system design is successful: the multiplexed feedback architecture is feasible, the protocol is functioning as designed, and the Piece-wise Linear Feedback method significantly compresses the data while does not affect the system's stability.

The proposed system followed the five approaches that discussed in Chapter 1 to reduce the system cost, and succeeded in each of them. The proposed system also provides better performance than existing designs. The thesis objectives are met.

7.2 Further Work

The future works can be performed in following aspects:

1. The proposed architecture does not include a self-start mechanism. The digital circuitry must be powered up by a standby power supply before it can be turned on. A self-start arrangement should be investigated. Specifically, the pulse-transformer can be arranged to deliver power.
2. Due to the over sampling strategy and the processing time, the duty cannot be updated before the next cycle begins. There are a few ways to solve this problem: (1) use two microcontrollers on the secondary side: one microcontroller does sampling and averaging, the other one implements encoder and transmitter; (2) use FPGA to implement digital circuitry; (3) apply the existing architecture on other topologies such as forward and full-bridge, which do not need over sampling strategy to get accurate samples.
3. The PLL is not implemented in the prototype. It can be added in the future.
4. The Flyback topology has a pole moving with load, which makes it difficult to compensate. With digital techniques, the load range can be divided into, say, 16 sections, and there are 16 corresponding PID algorithms stored in the primary controllers. The 4-bit load information can be sent with the voltage samples, so that the PID controller can dynamically choose compensation algorithms for corresponding load range.
5. The proposed architecture must be implemented into ASIC before it can actually save cost for manufacturers.

Appendix A PCB Layout

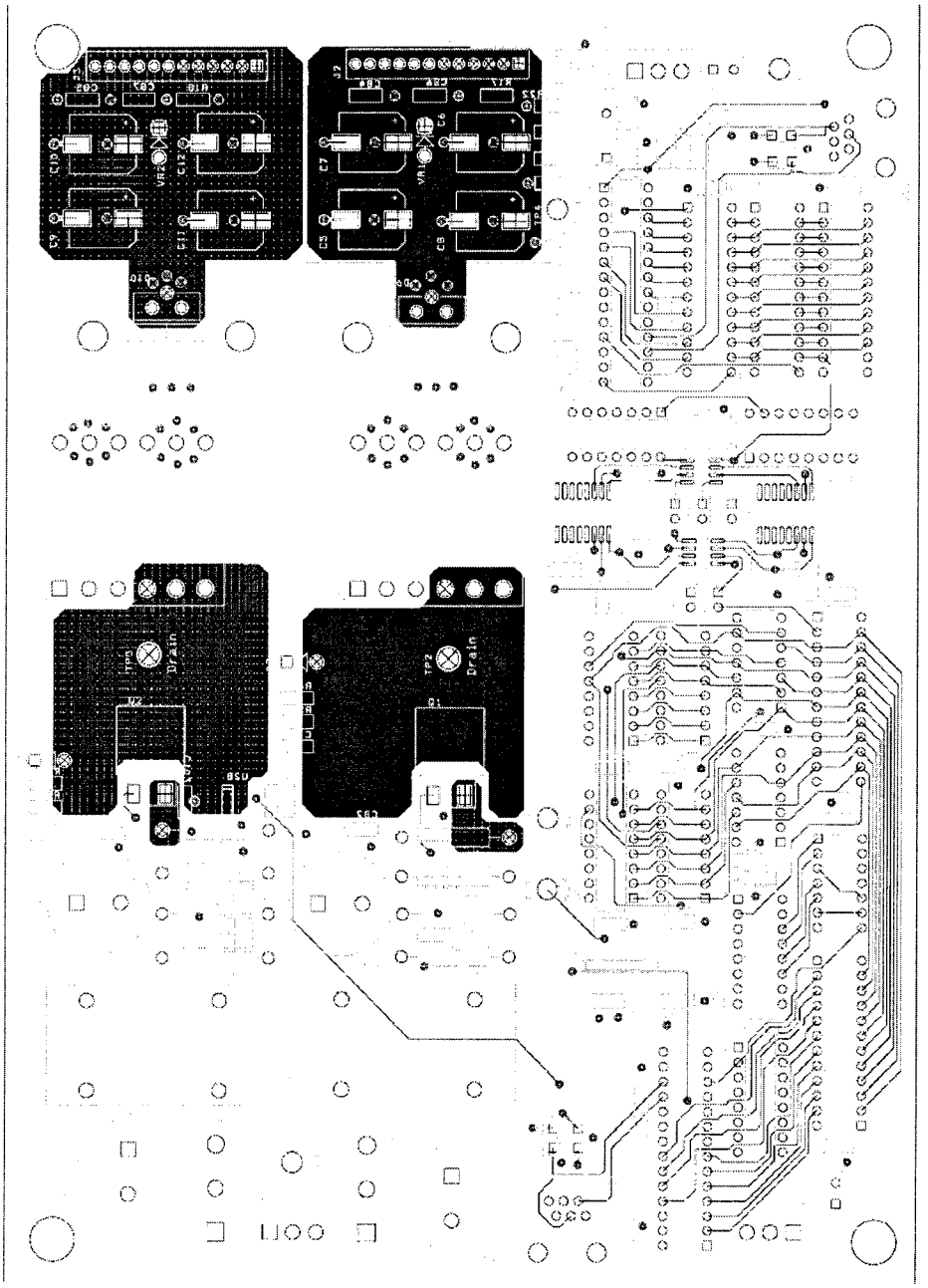


Figure A-1 TOP Layer with Silk Screen layers

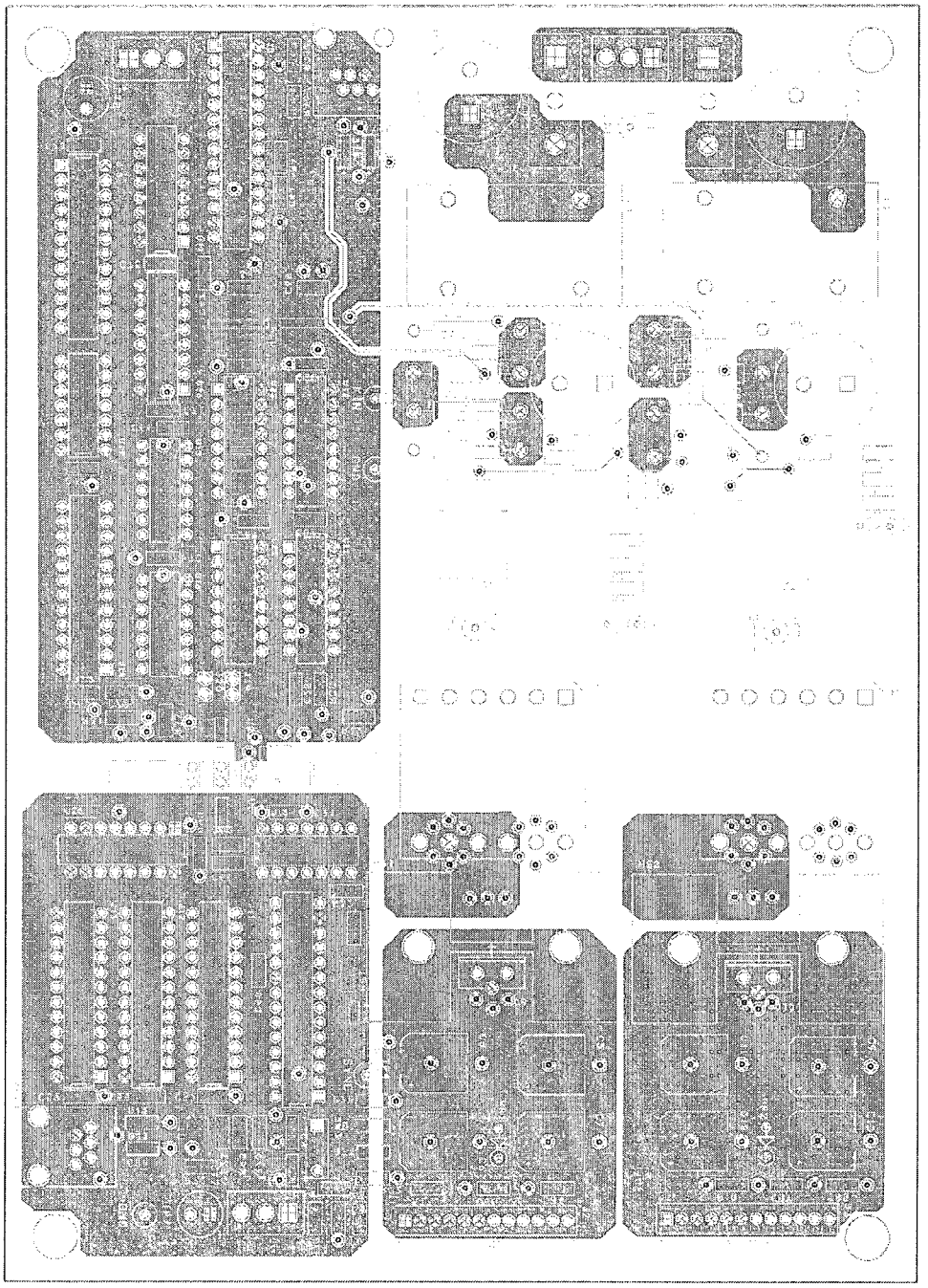


Figure A-2 POWER Layer with Silk Screen layers

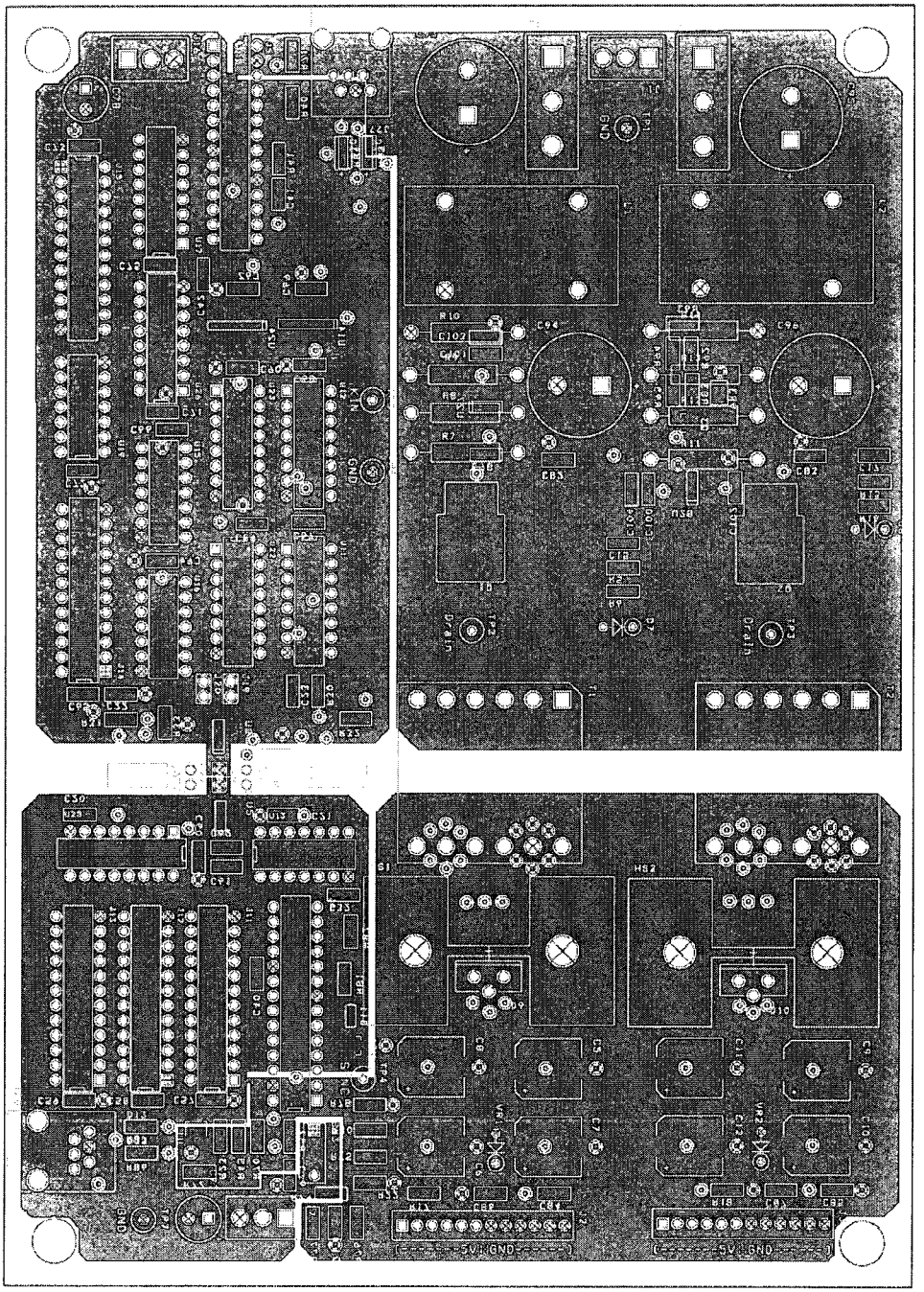


Figure A-3 GROUND Layer with Silk Screen layers

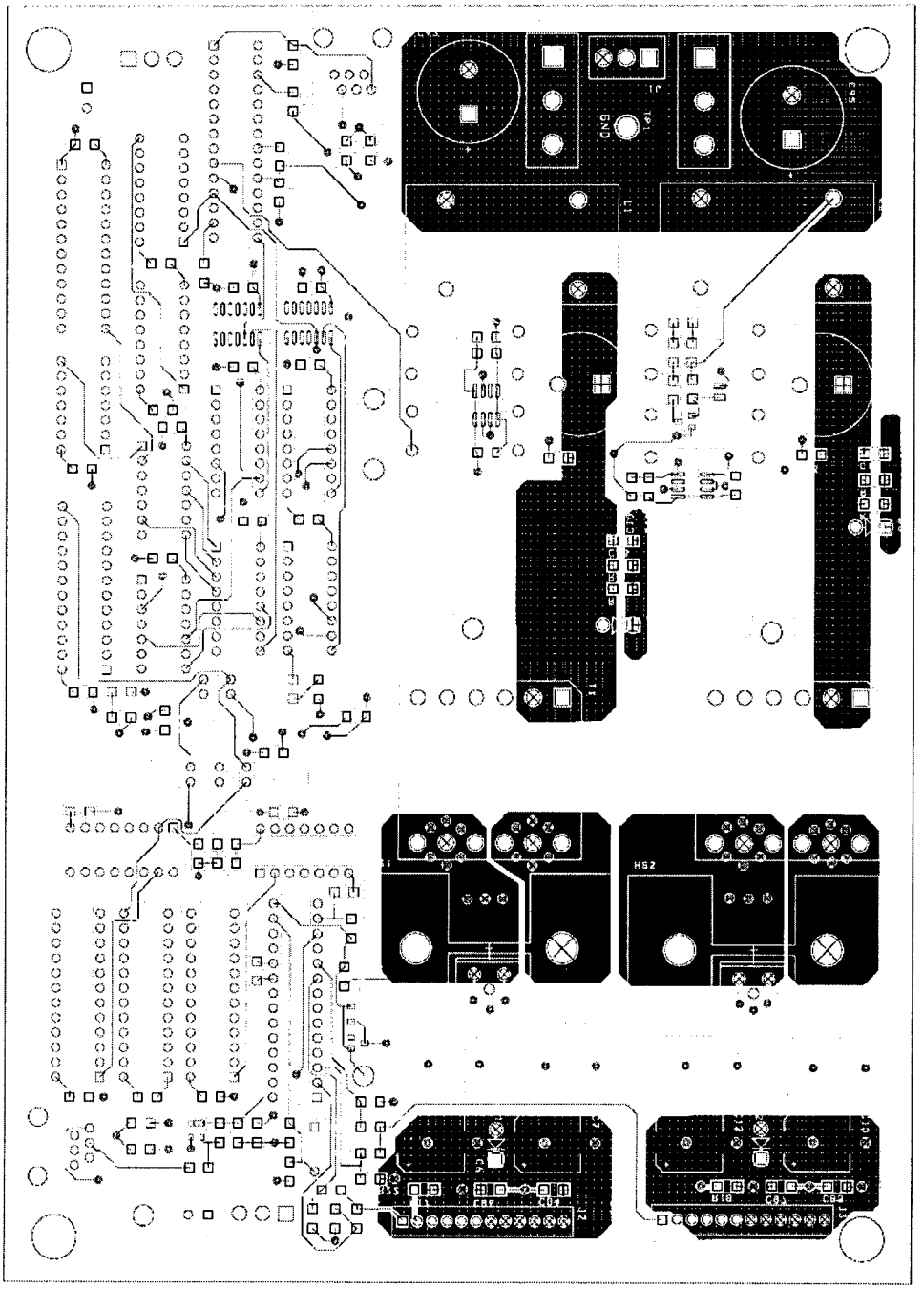


Figure A-4 BOTTOM Layer with Silk Screen layer

Appendix B Microcontroller Source Code

```
;*****************************************************************************
;                                                                             *
;   Filename        : Primary-side Code                                       *
;                                                                             *
;*****************************************************************************

    .equ __30F2020, 1
    .include "p30f2020.inc"

;.....
;Configuration bits:
;.....

    config __FOSC, CSW_FSCM_OFF & FRC_HI_RANGE & OSC2_IO

                        ;Turn off clock switching and
                        ;fail-safe clock monitoring and
                        ;use the XT osc and 4x PLL as
                        ;system clock
    config __FOSCSEL,FRC_PLL

    config __FPOR, PWRT_128
                        ; Set Brown-out Reset voltage and
                        ; and set Power-up Timer to 16msecs

    config __FGS, CODE_PROT_OFF     ;Set Code Protection Off for the
                        ; General Segment
.section My_Var,bss,near

SysSta:     .space 2    ; System State
Channel:    .space 2    ; Channel State
Table4to16: .space 32   ; 4B/16B LUT
TableHead:  .space 2    ; Store the head address of the 10B/8B LUT
Ik1: .space 4           ; Integrator for PWM1
Ik2: .space 4           ; Integrator for PWM2
Count: .space 2         ; Soft start counter
LastEk1: .space 2       ; Store last Ek for differential control
LastEk2: .space 2       ; Store last Ek for differential control


.text                   ;Start of Code section


.global __DefaultInterrupt
__DefaultInterrupt:
            bclr PORTF,#RF7
            bset PORTF,#RF8         ; turn LED1off,LED2on
                clrwdt
                bra $-2             ; dead loop
            reset ; reset the device

.global __PWM1Interrupt
__PWM1Interrupt:    ; "Stop Code Received" interrupt, the hardware will turn off PWM1 before entering this interrupt
                    bclr PTCON,#PTEN ;turn off PWM
                    bclr IFS1,#PWM1IF ;clear PWM1interrupt flag
                    bclr IEC1,#PWM1IE ;disable PWM1interrupt
                    clr SysSta      ; set System State to 00
                  ; bset PORTF,#RF7
                  ; bset PORTF,#RF8      ;turnLED1off,LED2 off
                    push w0
```

```
                mov #0x55,w0
                mov  w0,PDC1        ; reset to min duty
                mov  w0,PDC3
                clr Ik1
                clr Ik1+2
                clr Ik2
                clr Ik2+2
                clr A
                clr B
                bclr PWMCON1,#FLTIEN        ;disable PWM fault interrupt, clear flag
                pop w0
                retfie

.global __CNInterrupt
__CNInterrupt:              ; "Sync Code received" interrupt
                btsc SysSta,#1      ;if System State is 00 or 01, skip next line
                bra  EnableInt2
                bset SysSta,#1
                bclr SysSta,#0                ;Set SysSta to 10
                bclr IFS1,#PWM1IF             ;clear PWM1 interrupt flag
                bset IEC1,#PWM1IE             ;enable PWM1interrupt
                bset  PWMCON1,#FLTIEN         ;enable PWM fault interrupt
                bset  PTCON,#PTEN             ;turn on PWM module
                push w0
                mov #0,w0                     ;clear integrators
                mov w0,Ik1
                mov w0,Ik1+2
                mov w0,Ik2
                mov w0,Ik2+2
                mov w0,ACCAL
                mov w0,ACCAH
                mov w0,ACCAU
                mov w0,ACCBL
                mov w0,ACCBH
                mov w0,ACCBU
                pop w0
                clr LastEk1
                clr LastEk2
        ;       bclr PORTF,#RF7
        ;       bclr PORTF,#RF8         ;initiation completed, turn on LED1 and LED2
EnableInt2:
                bclr IFS1,#CNIF       ;clear CN interrupt flag
                bclr  IFS1,#INT2IF            ;clear INT2 interrupt flag
                bset  IEC1,#INT2IE            ;enable INT2 interrupt, prepare to receive AD sample
                retfie

.global __INT2Interrupt
__INT2Interrupt:    ; "Data Arrived" interrupt

                bclr  IFS1,#INT2IF
                bclr  IEC1,#INT2IE
                push.s
                ;read the ports
                mov PORTE,w0
                sl w0,#6,w0
                MOV #0b111110000000,w1
                and w0,w1,w1
                rlc PORTB,wreg
                MOV #0b1111110,w2
                and w2,w0,w0
                ior w0,w1,w2          ;move the result to w2 as LUT offset

                mov #0b00111110100,w0    ;test if it's the "Sync" code
                cp w2,w0
                bra z,Retn
                mov #0b11000001010,w0
                cp w2,w0
                bra z,Retn
```

```
                ;10B/8B decoder
                mov TableHead,w1
                add w2,w1,w0
                mov [w0], w1

                ;split 8b into 2 4b codes
                mov #0b1111,w0
                and w0,w1,w8        ;move 4B to W8 (PWM2)
                lsr w1,#4,w1
                and w0,w1,w1        ;move 4B to W1 (PWM1)

                ; exam the System State
                mov #0b10,w0
                mov SysSta,w3
                cpseq w0,w3
                bra  Pid                                    ;if not in soft start, go to PID
                ;if in soft start
                mov #0b1000,w0
                cp w1,w0
                bra gtu,StartOver
                cp w8,w0
                bra ltu,IncPdc

                ;before complete soft start
StartOver:

                mov #0b11,w0
                mov w0,SysSta

                mov     PDC1,w4
                mov #1,w5
                MPY w4*w5,A
                SFTAC A,#0
                mov ACCAL,w0            ;initiate integrator
                mov w0,Ik1
                mov ACCAH,w0
                mov w0,Ik1+2
                mov     PDC3,w4
                mov #1,w5
                MPY w4*w5,A
                mov ACCAL,w0            ; initiate integrator
                mov w0,Ik2
                mov ACCAH,w0
                bclr  IFS1,#INT2IF
                bclr  IEC1,#INT2IE
                pop.s
                retfie
;if Sync code, don't disable this interrupt, return
Retn:   bclr  IFS1,#INT2IF
                pop.s
                retfie
;soft start
IncPdc: inc Count
                mov #2,w0
                cp Count
                bra ltu,Retern
                clr Count
                mov #0xaff,w0
                cp PDC1
                bra gtu,Retern
                inc PDC1
                inc PDC3
Retern:
                bclr  IFS1,#INT2IF
                bclr  IEC1,#INT2IE
                pop.s
```

```
                retfie

;PID for PWM1
Pid:
                sl w1,w1
                mov #Table4to16,w0
                add w0,w1,w0
                mov [w0],w4
                mov Ik1+2,w6
                mov #1, w7
                MPY w6*w7,A
                SFTAC A,#-16
                mov #0xFFFF,w0
                cpsne w6,w0
                clr A
                mov Ik1,w6
                lsr w6,w6
                mov #2,w7
                btsc Ik1+2,#15
                bset w6,#15
                MPY w6*w7,B
                ADD     A
                SFTAC A,#-1
                mov #30,w5 ; Kp 24 746/8/4
                MAC W4*W5,A
                mov LastEk1,w1
                mov w4,LastEk1
                sub w4,w1,w5
                mov #5,w6 ;Kd
                MAC w5*w6,A
                BTSS ACCAH,#15
                bra Positi
                mov #0x55,w0
                mov w0,PDC1
                bra NewIk1
Positi:
                cp0 ACCAH
                bra nz,TooBig
                mov #0x8cf,w0
                cp ACCAL
                bra gtu,TooBig
                mov #0x55,w0
                cp ACCAL
                bra gtu,NC1
                mov w0,ACCAL
 NC1:
                mov ACCAL,w0
                mov w0,PDC1
                bra NewIk1
TooBig:
                mov     #0x8cf, w0
                mov w0,PDC1
NewIk1:
                mov Ik1+2, w0
                lac w0,     B
                mov Ik1, w0
                mov w0,   ACCBL
                mov     #1, w5
                MAC W4*W5,B
                sac B,w0
                mov w0,Ik1+2
                mov ACCBL,w0
                mov w0,Ik1


        ;PID for PWM2
                sl w8,w1
                mov #Table4to16,w0
```

```
                    add w0,w1,w0
                    mov [w0],w4
                    mov Ik2+2,w6
                    mov #1, w7
                    MPY w6*w7,A
                    SFTAC A,#-16
                    mov #0xFFFF,w0
                    cpsne w6,w0
                    clr A
                    mov Ik2,w6
                    lsr w6,w6
                    mov #2,w7
                    btsc Ik2+2,#15
                    bset w6,#15
                    MPY w6*w7,B
                    ADD     A
                    SFTAC A,#0
                    mov #30,w5
                    MAC W4*W5,A

                    mov LastEk2,w1
                    mov w4,LastEk2
                    sub w4,w1,w5
                    mov #5,w6 ;Kd
                    MAC w5*w6,A

                    BTSS ACCAH,#15
                    bra Positi2
                    mov     #0x55, w0
                    mov w0,PDC3
                    bra NewIk2
Positi2:
                    cp0 ACCAH
                    bra nz,TooBig2
                    mov #0x8cf,w0
                    cp ACCAL
                    bra gtu,TooBig2
                    mov #0x55,w0
                    cp ACCAL
                    bra gtu,NC2
                    mov w0,ACCAL

 NC2:
                     mov ACCAL,w0
                    mov w0,PDC3
                    bra NewIk2
TooBig2:
                    mov #0x8cf, w0
                    mov w0,PDC3

NewIk2:   mov Ik2+2, w0
                    lac w0,     B
                    mov Ik2, w0
                    mov w0,   ACCBL
                    mov     #1, w5
                    MAC W4*W5,B
                    sac B,w0
                    mov w0,Ik2+2
                    mov ACCBL,w0
                    mov w0,Ik2

                    pop.s
                    retfie

.global     __reset
__reset:
    MOV  #__SP_init, W15    ;Initalize the Stack Pointer
    MOV  #__SPLIM_init, W0  ;Initialize the Stack Pointer Limit Register
```

```
    MOV   W0, SPLIM
    NOP                         ;Add NOP to follow SPLIM initialization
    MOV #0x00E5,W0              ;initiate DSP
    MOV W0,CORCON
    mov #psvpage(TenToFourTable),w0 ;8 MSB of head address of 10b/8b LUT
    mov w0,PSVPAG
    mov #psvoffset(TenToFourTable),w1    ; 16 LSB of 10b/8b LUT head address
                mov w1,TableHead
    call   Initial                              ;Call the routine to setup I/O and PWM

;-------------------------------------------------------------------------------
; Main
;-------------------------------------------------------------------------------
CK_Start:
        clrwdt
        bra  CK_Start

;-------------------------------------------------------------------------------
; PWM and ADC setup code
;-------------------------------------------------------------------------------

Initial:
    mov   #7,w0
    mov   w0,OSCTUN
    CALL _wreg_init
    call Init_PORT
    call Init_PWM
    call Init_CN1
    call Init_INT2
    call Init_Table
    clr  SysSta
    bset SysSta,#0
    clr  Channel
    clr Count
    return


Init_Table:                     ;4b/16b LUT
                mov #0b0000000000110010,w0
                mov w0,Table4to16
                mov #0b0000000000100110,w0
                mov w0,Table4to16+2
                mov #0b0000000000011101,w0
                mov w0,Table4to16+4
                mov #0b0000000000010101,w0
                mov w0,Table4to16+6
                mov #0b0000000000001101,w0
                mov w0,Table4to16+8
                mov #0b0000000000000110,w0
                mov w0,Table4to16+10
                mov #0b0000000000000010,w0
                mov w0,Table4to16+12
                mov #0b0000000000000001,w0
                mov w0,Table4to16+14
                mov #0b0000000000000000,w0
                mov w0,Table4to16+16
                mov #0b1111111111111111,w0
                mov w0,Table4to16+18
                mov #0b1111111111111110,w0
                mov w0,Table4to16+20
                mov #0b1111111111111010,w0
                mov w0,Table4to16+22
                mov #0b1111111111110011,w0
                mov w0,Table4to16+24
                mov #0b1111111111101011,w0
                mov w0,Table4to16+26
                mov #0b1111111111100011,w0
                mov w0,Table4to16+28
```

```
                mov #0b1111111111011010,w0
                mov w0,Table4to16+30
                mov #0,w0
                mov w0,Ik1
                mov w0,Ik1+2
                mov w0,Ik2
                mov w0,Ik2+2
                return

Init_PORT:
                 mov            #0xFFFF,W0
                mov             w0,ADPCFG
                nop
                mov             W0,TRISA        ;All RA Ports input
                mov             W0,TRISB        ;All RB ports input
                mov             W0,TRISD        ;All RD Ports input，(RD0/IC1) input
                mov             #0xFFde,W0
                mov             W0,TRISE        ;RE0(PWM1L),RE5(PWM3H)output,Other input
                mov             #0xFE7F,W0
                mov             W0,TRISF        ;RF7 RF8 Output,Other Ports inputs (LEDs)
                mov             #0x0,W0
                mov             W0,PORTA
                mov             W0,PORTB
                mov             W0,PORTD
                mov             W0,PORTE
                mov             W0,PORTF        ;Clear all ports ;

                ; Now, ensure the power module is reset by driving the reset line for
                ; a few usec.
                return

Init_INT2:
                bset  INTCON2,#INT2EP
                bset  IPC4,#INT2IP0
                bclr  IPC4,#INT2IP1
                bset  IPC4,#INT2IP2
                bclr  IFS1,#INT2IF
                bclr  IEC1,#INT2IE
                return

Init_PWM:
    mov #0, w0
    mov w0, PTCON
    mov #0x2580, w0         ; Select period to be approximately 0.1usec
    mov w0, PTPER           ;period= (PTPER + 1) / 120 (30MIPS, unit:us)
    mov #55, w0
    mov w0, PDC1            ; min duty
    ; Code for PWM Output Control
    mov #0x4400, w0         ; PWM1L is controlled by PWM module
    mov w0, IOCON1          ;
    mov #55, w0
    mov w0, PDC3
    ; Code for PWM Output Control
    mov #0x8400, w0         ; PWM3H is controlled by PWM module
    mov w0, IOCON3
    mov #0x0440, w0
    mov w0, FCLCON1
    mov #0x0640, w0
    mov w0, FCLCON3

        mov  #0x1080,  w0
        mov  w0, PWMCON1

        mov  #0x0080,  w0
        mov  w0, PWMCON3
        bset  IPC4,#PWM1IP0
        bset  IPC4,#PWM1IP1
        bset  IPC4,#PWM1IP2
```

```
;           mov  #0x8080,w0
;           mov  w0,CMPCON3
;           mov  #0x0029,w0         ;(CMPREF * INTREF/1024)
;           mov  w0,CMPDAC3

;           mov  #0x8040,w0
;           mov  w0,CMPCON4
;           mov  #0x0028,w0
;           mov  w0,CMPDAC4          ;(CMPREF * INTREF/1024)

            return;


Init_CN1:
    bset CNPU1,#CN1PUE      ;enable weak pull up
    mov #0x6000, w0
    mov w0,IPC6
    bclr IFS1,#CNIF
    bset CNEN1,#CN1IE
    bset IEC1,#CNIE
    return                  ; return from the Setup routine

;..............................................................................
;Subroutine: Initialization of W registers to w0
;..............................................................................
_wreg_init:
    CLR W0
    MOV W0, W14
    REPEAT #12
    MOV W0, [++W14]
    CLR W14
;               clr temp
;               clr temp2
    RETURN

TenToFourTable:    .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b01001000   .word 0b00000000   .word 0b01000100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b11001000   .word 0b00000000   .word 0b11000100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00101000   .word 0b00000000   .word 0b00100100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b10101000   .word 0b00000000   .word 0b10100100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00001000   .word 0b00000000   .word 0b00000100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b01101000   .word 0b00000000   .word 0b01100100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b10001000   .word 0b00000000   .word 0b10000100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b11101000   .word 0b00000000   .word 0b11100100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b11100111   .word 0b00000000   .word 0b11110100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b10000111   .word 0b00000000   .word 0b10010100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b01100111   .word 0b00000000   .word 0b01110100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000111   .word 0b00000000   .word 0b00010100
.word 0b00000000   .word 0b00000000   .word 0b01010111   .word 0b01000111   .word 0b01011011   .word 0b01010100
.word 0b00000000   .word 0b00000000   .word 0b11010111   .word 0b11000111   .word 0b11011011   .word 0b11010100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b11110100
.word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000   .word 0b00000000
.word 0b00000000   .word 0b00000000   .word 0b00110111   .word 0b00100111   .word 0b00111011   .word 0b00110100
.word 0b00000000   .word 0b00000000   .word 0b10110111   .word 0b10100111   .word 0b10111011   .word 0b10110100
.word 0b00000000   .word 0b00000000   .word 0b00010111   .word 0b00000000   .word 0b00011011   .word 0b00010100
.word 0b00000000   .word 0b00000000   .word 0b01110111   .word 0b00000000   .word 0b01111011   .word 0b01110100
.word 0b00000000   .word 0b00000000   .word 0b10010111   .word 0b00000000   .word 0b10011011   .word 0b10010100
.word 0b00000000   .word 0b00000000   .word 0b11110111   .word 0b00000000   .word 0b11111011   .word 0b00000000
```

```
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b01100000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00111111  .word 0b10000000  .word 0b11100001  .word 0b01011110
.word 0b00000000  .word 0b00000000  .word 0b10111111  .word 0b11100000  .word 0b10000001  .word 0b11011110
.word 0b00000000  .word 0b00000000  .word 0b00011111  .word 0b00000000  .word 0b01100001  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b01111111  .word 0b00000000  .word 0b00000001  .word 0b00000000
.word 0b01011000  .word 0b00000000  .word 0b10011111  .word 0b11100110  .word 0b01000001  .word 0b00111110
.word 0b11011000  .word 0b00000000  .word 0b11111111  .word 0b10000110  .word 0b11000001  .word 0b10111110
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b01100110  .word 0b00000000  .word 0b00011110
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000110  .word 0b00000000  .word 0b01111110
.word 0b00111000  .word 0b00000000  .word 0b11101010  .word 0b01000110  .word 0b00100001  .word 0b10011110
.word 0b10111000  .word 0b00000000  .word 0b10001010  .word 0b11000110  .word 0b10100001  .word 0b11111110
.word 0b00011000  .word 0b00000000  .word 0b01101010  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01111000  .word 0b00000000  .word 0b00001010  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b10011000  .word 0b00000000  .word 0b01001010  .word 0b00100110  .word 0b00000000  .word 0b00000000
.word 0b11111000  .word 0b00000000  .word 0b11001010  .word 0b10100110  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000110  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b01100110  .word 0b00000000  .word 0b00000000
.word 0b11101100  .word 0b01011101  .word 0b00101010  .word 0b10000110  .word 0b11111110  .word 0b01000001
.word 0b10001100  .word 0b11011101  .word 0b10101010  .word 0b11100110  .word 0b10011110  .word 0b11000001
.word 0b01101100  .word 0b00000000  .word 0b00001010  .word 0b00000000  .word 0b01111110  .word 0b00000000
.word 0b00001100  .word 0b00000000  .word 0b01101010  .word 0b00000000  .word 0b00011110  .word 0b00000000
.word 0b01001100  .word 0b00111101  .word 0b10001010  .word 0b11110110  .word 0b01011110  .word 0b00100001
.word 0b11001100  .word 0b10111101  .word 0b11101010  .word 0b10010110  .word 0b11011110  .word 0b10100001
.word 0b00000000  .word 0b00011101  .word 0b00000000  .word 0b01110110  .word 0b00000000  .word 0b00000001
.word 0b00000000  .word 0b01111101  .word 0b00000000  .word 0b00010110  .word 0b00000000  .word 0b01100001
.word 0b00101100  .word 0b10011101  .word 0b11111010  .word 0b01010110  .word 0b00111110  .word 0b10000001
.word 0b10101100  .word 0b11111101  .word 0b10011010  .word 0b11010110  .word 0b10111110  .word 0b11100001
.word 0b00001100  .word 0b00000000  .word 0b01111010  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01101100  .word 0b00000000  .word 0b00011010  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b10001100  .word 0b00000000  .word 0b01011010  .word 0b00110110  .word 0b00000000  .word 0b11110001
.word 0b11101100  .word 0b00000000  .word 0b11011010  .word 0b10110110  .word 0b00000000  .word 0b10010001
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00010110  .word 0b00000000  .word 0b01110001
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b01110110  .word 0b00000000  .word 0b00010001
.word 0b11111100  .word 0b01000010  .word 0b00111010  .word 0b10010110  .word 0b00000000  .word 0b01010001
.word 0b10011100  .word 0b11000010  .word 0b10111010  .word 0b11110110  .word 0b00000000  .word 0b11010001
.word 0b01111100  .word 0b00000000  .word 0b00011010  .word 0b00000000  .word 0b00000000  .word 0b11110001
.word 0b00011100  .word 0b00000000  .word 0b01111010  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01011100  .word 0b00100010  .word 0b10011010  .word 0b11110000  .word 0b00000000  .word 0b00110001
.word 0b11011100  .word 0b10100010  .word 0b11111010  .word 0b10010000  .word 0b00000000  .word 0b10110001
.word 0b00000000  .word 0b00000010  .word 0b00000000  .word 0b01110000  .word 0b00000000  .word 0b00010001
.word 0b00000000  .word 0b01100010  .word 0b00000000  .word 0b00010000  .word 0b00000000  .word 0b01110001
.word 0b00111100  .word 0b10000010  .word 0b11101111  .word 0b01010000  .word 0b00000000  .word 0b10010001
.word 0b10111100  .word 0b11100010  .word 0b10001111  .word 0b11010000  .word 0b00000000  .word 0b00000000
.word 0b00011100  .word 0b00000000  .word 0b01101111  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01111100  .word 0b00000000  .word 0b00001111  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b10011100  .word 0b11110010  .word 0b01001111  .word 0b00110000  .word 0b00000000  .word 0b00000000
.word 0b11111100  .word 0b10010010  .word 0b11001111  .word 0b10110000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01110010  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00010010  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01010010  .word 0b00101111  .word 0b00000000  .word 0b00000000  .word 0b01010000
.word 0b00000000  .word 0b11010010  .word 0b10101111  .word 0b00000000  .word 0b00000000  .word 0b11010000
.word 0b00000000  .word 0b11110010  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00110010  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00110000
.word 0b00000000  .word 0b10110010  .word 0b00000000  .word 0b10001110  .word 0b00000000  .word 0b10110000
.word 0b00000000  .word 0b00010010  .word 0b00000000  .word 0b01101110  .word 0b00000000  .word 0b00010000
.word 0b00000000  .word 0b01110010  .word 0b00000000  .word 0b00001110  .word 0b00000000  .word 0b01110000
.word 0b00000000  .word 0b10010010  .word 0b00000000  .word 0b01001110  .word 0b00000000  .word 0b10010000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b11001110  .word 0b00000000  .word 0b11110000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b11101110  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b01000000  .word 0b00101110  .word 0b00000000  .word 0b11101001
.word 0b00000000  .word 0b00000000  .word 0b11000000  .word 0b10101110  .word 0b00000000  .word 0b10001001
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00001110  .word 0b00000000  .word 0b01101001
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b01101110  .word 0b00000000  .word 0b00001001
.word 0b00000000  .word 0b01011111  .word 0b00100000  .word 0b10001110  .word 0b00000000  .word 0b01001001
.word 0b00000000  .word 0b11011111  .word 0b10100000  .word 0b11101110  .word 0b00000000  .word 0b11001001
```

```
.word 0b00000000  .word 0b00000101  .word 0b00000000  .word 0b01110011  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01100101  .word 0b00000000  .word 0b00010011  .word 0b00000000  .word 0b00000000
.word 0b00101001  .word 0b10000101  .word 0b11111101  .word 0b01010011  .word 0b00111011  .word 0b00000000
.word 0b10101001  .word 0b11100101  .word 0b10011101  .word 0b11010011  .word 0b10111011  .word 0b00000000
.word 0b00001001  .word 0b00000000  .word 0b01111101  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01101001  .word 0b00000000  .word 0b00011101  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b10001001  .word 0b11110101  .word 0b01011101  .word 0b00110011  .word 0b00000000  .word 0b00000000
.word 0b11101001  .word 0b10010101  .word 0b11011101  .word 0b10110011  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01110101  .word 0b00000000  .word 0b00010011  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00010101  .word 0b00000000  .word 0b01110011  .word 0b00000000  .word 0b00000000
.word 0b11111001  .word 0b01010101  .word 0b00111101  .word 0b10010011  .word 0b00000000  .word 0b00000000
.word 0b10011001  .word 0b11010101  .word 0b10111101  .word 0b11110011  .word 0b00000000  .word 0b00000000
.word 0b01111001  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00011001  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01011001  .word 0b00110101  .word 0b00000000  .word 0b11111000  .word 0b00000000  .word 0b00000000
.word 0b11011001  .word 0b10110101  .word 0b00000000  .word 0b10011000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00010101  .word 0b00000000  .word 0b01111000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01110101  .word 0b00000000  .word 0b00011000  .word 0b00000000  .word 0b00000000
.word 0b00111001  .word 0b10010101  .word 0b00000000  .word 0b01011000  .word 0b00000000  .word 0b00000000
.word 0b10111001  .word 0b11110101  .word 0b00000000  .word 0b11011000  .word 0b00000000  .word 0b00000000
.word 0b00011001  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01111001  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b10011001  .word 0b11111111  .word 0b00000000  .word 0b00111000  .word 0b00000000  .word 0b00000000
.word 0b11111001  .word 0b10011111  .word 0b00000000  .word 0b10111000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01111111  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00011111  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b11100000  .word 0b01011111  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b10000000  .word 0b11011111  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01100000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01000000  .word 0b00111111  .word 0b00000000  .word 0b00000000  .word 0b01000111  .word 0b00000000
.word 0b11000000  .word 0b10111111  .word 0b00000000  .word 0b10001011  .word 0b11000111  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b01101011  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00001011  .word 0b00000000  .word 0b00000000
.word 0b00100000  .word 0b00000000  .word 0b00000000  .word 0b01001011  .word 0b00100111  .word 0b00000000
.word 0b10100000  .word 0b00000000  .word 0b00000000  .word 0b11001011  .word 0b10100111  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000111  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b11101011  .word 0b01100111  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00101011  .word 0b10000111  .word 0b00000000
.word 0b00000000  .word 0b10001101  .word 0b00000000  .word 0b10101011  .word 0b11100111  .word 0b00000000
.word 0b00000000  .word 0b01101101  .word 0b00000000  .word 0b00001011  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00001101  .word 0b00000000  .word 0b01101011  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01001101  .word 0b00000000  .word 0b10001011  .word 0b11101000  .word 0b00000000
.word 0b00000000  .word 0b11001101  .word 0b00000000  .word 0b11101011  .word 0b10001000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b01101000  .word 0b00000000
.word 0b00000000  .word 0b11101101  .word 0b00000000  .word 0b00000000  .word 0b00001000  .word 0b00000000
.word 0b01001111  .word 0b00101101  .word 0b00000000  .word 0b11100100  .word 0b01001000  .word 0b00000000
.word 0b11001111  .word 0b10101101  .word 0b00000000  .word 0b10000100  .word 0b11001000  .word 0b00000000
.word 0b00000000  .word 0b00001101  .word 0b00000000  .word 0b01100100  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01101101  .word 0b00000000  .word 0b00000100  .word 0b00000000  .word 0b00000000
.word 0b00101111  .word 0b10001101  .word 0b11100011  .word 0b01000100  .word 0b00101000  .word 0b00000000
.word 0b10101111  .word 0b11101101  .word 0b10000011  .word 0b11000100  .word 0b10101000  .word 0b00000000
.word 0b00001111  .word 0b00000000  .word 0b01100011  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b01101111  .word 0b00000000  .word 0b00000011  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b10001111  .word 0b11100010  .word 0b01000011  .word 0b00100100  .word 0b00000000  .word 0b00000000
.word 0b11101111  .word 0b10000010  .word 0b11000011  .word 0b10100100  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b01100010  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000010  .word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00000000
.word 0b11100101  .word 0b01000010  .word 0b00100011  .word 0b00000000  .word 0b11110111  .word 0b00000000
.word 0b10000101  .word 0b11000010  .word 0b10100011  .word 0b00000000  .word 0b10010111  .word 0b00000000
.word 0b01100101  .word 0b00000000  .word 0b00000011  .word 0b00000000  .word 0b01110111  .word 0b00000000
.word 0b00000101  .word 0b00000000  .word 0b01100011  .word 0b00000000  .word 0b00010111  .word 0b00000000
.word 0b01000101  .word 0b00100010  .word 0b10000011  .word 0b11111011  .word 0b01010111  .word 0b00000000
.word 0b11000101  .word 0b10100010  .word 0b11100011  .word 0b10011011  .word 0b11010111  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b01111011  .word 0b00000000  .word 0b00000000
.word 0b00000000  .word 0b00000000  .word 0b00000000  .word 0b00011011  .word 0b00000000  .word 0b00000000
.word 0b00100101  .word 0b00000000  .word 0b11110011  .word 0b01011011  .word 0b00110111  .word 0b00000000
.word 0b10100101  .word 0b00000000  .word 0b10010011  .word 0b11011011  .word 0b10110111  .word 0b00000000
```

```
.word 0b00000000    .word 0b00000000    .word 0b00000000    .word 0b00000000    .word 0b00000000
.word 0b00000000    .word 0b00000000    .word 0b00000000    .word 0b00000000    .end
.word 0b00000000    .word 0b00000000    .word 0b00000000    .word 0b00000000    ;End of program code
.word 0b00000000    .word 0b00000000    .word 0b00000000    .word 0b00000000
```

```
;*****************************************************************************
;                                                                            *
;   Filename            : Secondary-side Code                                *
;                                                                            *
;*****************************************************************************

      .equ __30F2020, 1
      .include "p30f2020.inc"

;.....
;Configuration bits:
;.....

      config __FOSC, CSW_FSCM_OFF & FRC_HI_RANGE &OSC2_IO    ;Turn off clock switching and
                                    ;fail-safe clock monitoring and
                                    ;use the XT osc and 4x PLL as
                                    ;system clock
      config __FOSCSEL,FRC_PLL

      config __FPOR, PWRT_128
                              ;Set Brown-out Reset voltage and
                              ;and set Power-up Timer to 16msecs

      config __FGS, CODE_PROT_OFF       ;Set Code Protection Off for the
                              ;General Segment
.equiv T1_Period,0x33;0x3c          ;2us

;.....
;Uninitialized variables in Near data memory (Lower 8Kb of RAM)
;.....
.section My_Var,bss,near;
Data:    .space 2
SysSta:  .space 2     ;System State
StopSta: .space 2     ;Stop State
KeyCnt:  .space 2     ;debouncing
Channel: .space 2     ;Channel state
Times4:     .space 2     ;ADC completed flag
AdCCnt:  .space 2     ;ADC counter
Rlt8b:   .space 2     ; 8bit result
Ad0Buf:  .space 2     ;AD0 buffer
Ad1Buf:  .space 2     ;AD1 buffer
AD0SUM:  .space 2     ;AD0 sum
AD1SUM:  .space 2     ;AD1 sum
Rlt10b1: .space 2          ;store 10b negative code
Rlt10b2: .space 2          ;store 10b positive code
SendCTL: .space 2          ;send number control word

.section CNjumpTable,bss          ; CN Jump Table
Branch0Adr: .space 4
Branch1Adr: .space 4
Branch2Adr: .space 4
Branch3Adr: .space 4

;.....
;Code Section in Program Memory
;.....

.text                         ;Start of Code section

.global      __reset
__reset:
      MOV               #__SP_init, W15     ;Initalize the Stack Pointer
      MOV               #__SPLIM_init, W0   ;Initialize the Stack Pointer Limit Register
      MOV               W0, SPLIM
      NOP                         ;Add NOP to follow SPLIM initialization
      call    Initial
```

```
;-------------------------------------------------------------------------
; Main
;-------------------------------------------------------------------------
CK_Start:
        clr  KeyCnt
Star_Key:
        clrwdt
        btss PORTA,#RA9
        inc  KeyCnt
        nop
        btsc PORTA,#RA9
        clr  KeyCnt
        mov  #60000,w0
        cp   KeyCnt
        bra  ltu,Star_Key
        clr  KeyCnt
        bset PTCON, #15

        CLR SysSta
        CLR StopSta
        CLR KeyCnt
        CLR Channel
        CLR Times4
        CLR AdCCnt
        CLR SendCTL
        mov         #0b0000000110001011,w0
        mov w0,Rlt10b1
        mov w0,Rlt10b2
        bclr IFS0,#IC1IF
        bset IEC0,#IC1IE
        bclr IFS1,#CNIF
        bset IEC1,#CNIE
        bclr SysSta,#0
        bset SysSta,#1
        bclr IFS1,#AC3IF
        bset IEC1,#AC3IE
        bclr IFS2,#AC4IF
        bset IEC2,#AC4IE

If_New:
        btss Times4,#0                  ;If AD completed, go on processing
        bra  Stop_Key                   ;Otherwise, go to Stop_Key
        mov    w4,PSVPAG
        rrnc AD0SUM,WREG
        bclr w0,#0
        bclr w0,#15
        add w5,w0,w0
        mov [w0], w2

        rrnc AD1SUM,WREG
        bclr w0,#0
        bclr w0,#15
        add w5,w0,w0
        mov [w0], w1

        sl w2,#0x4,w2
        ior w2,w1,w0
        mov w0,Rlt8b

         mov  w8,PSVPAG
        rlnc Rlt8b,WREG
        add w0,w9,w0
        mov [w0], w1
        mov w1,Rlt10b1

        mov w6,PSVPAG
        rlnc Rlt8b,WREG
        add w0,w7,w0
```

```
        mov [w0], w1
        mov w1,Rlt10b2

        clr Times4
        btss StopSta,#0
        clr SendCTL
        bclr IFS1,#CNIF
        bset IEC1,#CNIE

Stop_Key:
        cp0 StopSta
        bra nz,Stop_Deal

        btss PORTA,#RA9
        bra If_New

        inc KeyCnt
        btss PORTA,#RA9
        clr  KeyCnt
        mov  #60000,w0
        cp   KeyCnt
        bra  ltu,Stop_Key

Stop_Deal:
        clr  KeyCnt
        mov #3,w0
        mov w0,SendCTL
        bset IEC1,#CNIE
        bset StopSta,#0
        clr T2CON
        clr TMR2
        mov #60000,w0
        mov w0,PR2
        bclr IFS0,#T2IF
        bset IEC0,#T2IE
        bclr IPC1,#T2IP0
        bset IPC1,#T2IP1
        bset IPC1,#T2IP2
        bset T2CON,#TON

SysSta0:
        clrwdt
        cp0 SysSta
        bra nz,SysSta0
        bclr PTCON, #15         ; turn off PWM module
        bclr IEC1,#CNIE
        bclr IFS1,#CNIF
        bclr IEC0,#IC1IE
        bclr IFS0,#IC1IF
        bclr T2CON,#TON

Stop_Key1:
        clrwdt
        btsc PORTA,#RA9
        inc  KeyCnt
        nop
        btss PORTA,#RA9
        clr  KeyCnt
        mov  #60000,w0
        cp   KeyCnt
        bra  ltu,Stop_Key1
        bra  CK_Start


.global __DefaultInterrupt
__DefaultInterrupt:
                bset PORTF,#RF7
                bclr PORTF,#RF8
```

```
                clrwdt
                bra $-2
                reset

.global __T2Interrupt
__T2Interrupt:          ;enter T2Interrupt after PWM shut down
                bclr IFS0,#T2IF
                clr  SysSta
                bclr T2CON,#TON
                retfie

.global __CMP3Interrupt
__CMP3Interrupt:        ;PWM2 overvoltage
        bclr IEC1,#AC3IE
        bclr IFS1,#AC3IF
        bset StopSta,#0
        mov #3,w0
        mov w0,SendCTL
        bclr IFS1,#CNIF
        bset IEC1,#CNIE
        retfie

.global __CMP4Interrupt
__CMP4Interrupt:        ;PWM1 overvoltage
        bclr IEC2,#AC4IE
        bclr IFS2,#AC4IF
        bset StopSta,#0
        mov #3,w0
        mov w0,SendCTL
        clr IFS1,#CNIF
        bset IEC1,#CNIE
        retfie

.global __ADCInterrupt
__ADCInterrupt:
                BCLR IFS0,#ADIF         ; Clear the AD interrupt flag
                BCLR ADSTAT, #0         ; Clear Pair 0
                push.w w0
                MOV ADCBUF0, w0         ; Read channel 0 conversion result
                add Ad0Buf
                MOV ADCBUF1, w0         ; Read channel 1 conversion result
                add Ad1Buf
                inc AdCCnt
                mov #3, w0
                cp  AdCCnt
                bra ltu,Retn
                bra z,Retn2
                mov Ad0Buf,W0
                MOV W0,AD0SUM
                mov Ad1Buf,W0
                MOV W0,AD1SUM
                bset Times4,#0
                clr  AdCCnt
Retn:
                pop.w  w0
                retfie
Retn2:
                bclr T1CON,#TON
                mov #T1_Period-1,w0
                mov w0,TMR1
                pop.w  w0
                retfie

.global __IC1Interrupt
__IC1Interrupt:                         ;PWM Synchronization interrupt
                btsc StopSta,#0
                bra  EnAdC
                bset SysSta,#0
```

```
                bset T1CON,#TON
                clr  AdCCnt
                clr Ad0Buf
                clr Ad1Buf
                bclr IFS0,#IC1IF
                retfie
EnAdC:
                bclr IFS0,#IC1IF
                clr TMR2
                retfie

.global __CNInterrupt
__CNInterrupt:                          ; Request Data interrupt
                PUSH.S
                mov SendCTL,w0
                sl  w0,#2,w0
                add w0,w10,w1
                mov [w1],w0
                goto w0

Branch0:
                btsc Channel,#0
                bra  SendHStart
                mov #0B0000000011111010,W0
                mov w0,PORTE
                mov #0B0000000000001100,W0
                mov w0,PORTB
                bset   PORTF,#RF6
                ;nop
                bclr   PORTF,#RF6
                bset Channel,#0
                bclr IFS1,#CNIF
                inc SendCTL
                POP.S
                 retfie
SendHStart:
                mov #0B0000001100000101,W0
                mov w0,PORTE
                mov #0B0000000000110000,W0
                mov w0,PORTB
                bset   PORTF,#RF6
                bclr   PORTF,#RF6
                bclr Channel,#0
                bclr IFS1,#CNIF
                inc SendCTL
                POP.S
                retfie

Branch1:
                btsc Channel,#0
                bra  CkHtable
                mov Rlt10b1,w1
                btsc w1,#15
                bset Channel,#0
                mov w1,PORTE
          1     sr w1,#0x4,w1
                mov w1,PORTB
                bset   PORTF,#RF6
                bclr   PORTF,#RF6
                bclr IFS1,#CNIF
                inc SendCTL
                POP.S
                retfie
CkHtable:
                mov Rlt10b2,w1
                btsc w1,#15
                bclr Channel,#0
                mov w1,PORTE
```

```
                lsr w1,#0x4,w1
                mov w1,PORTB
                bset   PORTF,#RF6
                bclr   PORTF,#RF6
                bclr IFS1,#CNIF
                inc SendCTL
                POP.S
                retfie

Branch2:
                btsc SysSta,#0
                bclr IEC1,#CNIE
                mov #0b0000000110001011,W0
                mov w0,PORTE
                mov #0B0000000000011000,W0
                mov w0,PORTB
                bset   PORTF,#RF6
                bclr   PORTF,#RF6
                bclr IFS1,#CNIF
                clr  SendCTL
                POP.S
                retfie

Branch3:

                btsc Channel,#0
                bra  SendHStop
                mov #0B0000000011111001,W0
                mov w0,PORTE
                mov #0B0000000000001100,W0
                mov w0,PORTB
                bset   PORTF,#RF6
                bclr   PORTF,#RF6
                bset Channel,#0
                bclr IFS1,#CNIF
                POP.S
                retfie
SendHStop:
                mov #0B0000001100000110,W0;
                mov w0,PORTE
                mov #0B0000000000110000,W0
                mov w0,PORTB
                bset   PORTF,#RF6
                bclr   PORTF,#RF6
                bclr Channel,#0
                bclr IFS1,#CNIF
                POP.S
                retfie


Initial:
        mov   #7,w0
        mov   w0,OSCTUN
        CALL _wreg_init
        call Init_PORT
        call Init_PWM
        call Init_CN0
        call Init_IC1
        call Init_AD
        call Init_T1
        call Init_CMP
        bset.b  CORCONL,#PSV            ; enable Program Space Visibility

                mov #psvpage(FTable),w4
                mov #psvoffset(FTable),w5
                mov #psvpage(HTable),w6
                mov #psvoffset(HTable),w7
                mov #psvpage(LTable),w8
```

```
                mov #psvoffset(LTable),w9
                bset SysSta,#0          ;状态字 01
                return

Init_T1: ; AD Timer
                CLR T1CON
                mov #T1_Period-1,w0
                mov w0,TMR1
                MOV #T1_Period,w0
                mov w0,PR1
                return
Init_PORT:
                mov             #0xFFFF,W0
                mov             W0,TRISA        ; All RA Ports input
                mov             #0xFFC3,W0
                mov             W0,TRISB        ;RB2,RB3,RB4,RB5 output,Other input
                mov             #0xFFFF,W0
                mov             W0,TRISD        ;All RD Ports input,
                mov             #0xFF40,W0
                mov             W0,TRISE        ;RE0,RE1,RE2,RE3,RE4,RE5,RE7(PWM4H)output,Other input
                mov             #0xFE3F,W0
                mov             W0,TRISF        ;RF6 RF7 RF8 Output,Other Ports input (LEDs)
                mov             #0x0,W0
                mov             W0,PORTA
                mov             W0,PORTB
                mov             W0,PORTD
                mov             W0,PORTE
                mov             W0,PORTF        ;Clear all ports ;

                mov   w0,Data
                mov   w0,AdCCnt
                ; Now, ensure the power module is reset by driving the reset line for
                ; a few usec.
                return
Init_PWM:
        mov #0x0000, w0
        mov w0, PTCON
        mov #0x078, w0          ; Select period to be approximately 0.1usec
        mov w0, PTPER           ; period = (PTPER + 1) / 120 (30MIPS unit:us)
        mov #0x3c, w0
        mov w0, PDC4            ; the PDC1 value = 0.5*(PWM Period)
    ; Code for PWM Output Control
        mov #0x8400, w0         ; PWM4H  is controlled by PWM module
        mov w0, IOCON4
        mov  #0x0080,  w0
        mov  w0, PWMCON4
        mov  #0x0003,  w0
        mov  w0,FCLCON4
        return;

Init_CMP:
        mov  #0x80e1,w0
        mov  w0,CMPCON3
         mov  #0x3f0,w0         ;(CMREF * INTREF/1024)
        mov  w0,CMPDAC3
        bclr IEC1,#AC3IE
        bclr IFS1,#AC3IF
        mov  #0x7000,w0
        mov  w0,IPC7
        mov  #0x8021,w0
        mov  w0,CMPCON4
        mov  #0x3f0,w0
        mov  w0,CMPDAC4         ;(CMREF * INTREF/1024)
        bclr IEC2,#AC4IE
        bclr IFS2,#AC4IF
        mov  #0x7,w0
        mov  w0,IPC8
        return
```

```
Init_CN0:
        bset CNPU1,#CN0PUE
        mov #0x5000, w0
        mov w0,IPC6
        bclr IFS1,#CNIF
        bset CNEN1,#CN0IE
        bclr IEC1,#CNIE

        MOV   #handle(Branch0), w0
        MOV   w0, Branch0Adr
        MOV   #handle(Branch1), w0
        MOV   w0, Branch1Adr
        MOV   #handle(Branch2), w0
        MOV   w0, Branch2Adr
        MOV   #handle(Branch3), w0
        MOV   w0, Branch3Adr
        MOV    #.startof.(CNjumpTable), w10
        return

Init_IC1:
        mov #0x02, w0
        mov w0,IC1CON
        bclr IFS0,#IC1IF
        bclr IEC0,#IC1IE
        bset IPC0,#IC1IP1
        bset IPC0,#IC1IP2
        bclr IPC0,#IC1IP0
        return

Init_AD:
        bclr   ADCON,#ADSIDL
        bclr   ADCON,#FORM
        bclr   ADCON,#EIE
        bclr   ADCON,#ORDER
        bset   ADCON,#SEQSAMP
        bclr   ADCON,#ADCS2
        bset   ADCON,#ADCS1
        bset   ADCON,#ADCS0      ; Clock Divider is set up for Fadc/10 Tad=41.6ns
        mov  #0xFFFC,W0
        mov   W0,ADPCFG          ; AN0 and AN1 are analog inputs
        mov   #0,W0
        mov   W0,ADSTAT          ; Clear the ADSTAT register
        mov   #0x8c,W0
        mov   W0,ADCPC0
        bclr   IFS0,#ADIF        ; Clear AD Interrupt Flag
        BSET  IPC2,#ADIP2
        BCLR  IPC2,#ADIP1
        Bclr   IPC2,#ADIP0
        BSET   IEC0,#ADIE        ; Enable the AD interrupt
        bset   ADCON,#ADON
        return
;.....
;Subroutine: Initialization of W registers to 0x0000
;.....
_wreg_init:
        CLR W0
        MOV W0, W14
        REPEAT #12
        MOV W0, [++W14]
        CLR W14
        CLR SysSta
        CLR StopSta
        CLR KeyCnt
        CLR Channel
        CLR Times4
        CLR AdCCnt
        RETURN
```

```
HTable:; 8b/10b RD+table
.word 0b0000000110001011
.word 0b0000001000101011
.word 0b0000000100101011
.word 0b1000001100010100
.word 0b0000000010101011
.word 0b1000001010010100
.word 0b1000000110010100
.word 0b1000000001110100
.word 0b0000000001101011
.word 0b1000001001010100
.word 0b1000000101010100
.word 0b1000001101000100
.word 0b1000000011010100
.word 0b1000001011000100
.word 0b1000000111000100
.word 0b0000001010001011
.word 0b0000001001001011
.word 0b1000001000110100
.word 0b1000000100110100
.word 0b1000001100100100
.word 0b1000000010110100
.word 0b1000001010100100
.word 0b1000000110100100
.word 0b0000000001011011
.word 0b0000000011001011
.word 0b1000001001100100
.word 0b1000000101100100
.word 0b0000000010011011
.word 0b1000000011100100
.word 0b0000000100011011
.word 0b0000001000011011
.word 0b0000000101001011
.word 0b1000000110001001
.word 0b1000001000101001
.word 0b1000000100101001
.word 0b0000001100011001
.word 0b1000000010101001
.word 0b0000001010011001
.word 0b0000000110011001
.word 0b0000000001111001
.word 0b1000000001101001
.word 0b0000001001011001
.word 0b0000000101011001
.word 0b0000001101001001
.word 0b0000000011011001
.word 0b0000001011001001
.word 0b0000000111001001
.word 0b1000001010001001
.word 0b1000001001001001
.word 0b0000001000111001
.word 0b0000000100111001
.word 0b0000001100101001
.word 0b0000000010111001
.word 0b0000001010101001
.word 0b0000000110101001
.word 0b1000000001011001
.word 0b1000000011001001
.word 0b0000001001101001
.word 0b0000000101101001
.word 0b1000000010011001
.word 0b0000000011101001
.word 0b1000000100011001
.word 0b1000001000011001
.word 0b1000000101001001
.word 0b1000000110000101
.word 0b1000001000100101
.word 0b1000000100100101
.word 0b0000001100010101
.word 0b1000000010100101
.word 0b0000001010010101
.word 0b0000000110010101
.word 0b0000000001110101
.word 0b1000000001100101
.word 0b0000001001010101
.word 0b0000000101010101
.word 0b0000001101000101
.word 0b0000000011010101
.word 0b0000001011000101
.word 0b0000000111000101
.word 0b1000001010000101
.word 0b1000001001000101
.word 0b0000001000110101
.word 0b0000000100110101
.word 0b0000001100100101
.word 0b0000000010110101
.word 0b0000001010100101
.word 0b0000000110100101
.word 0b1000000001010101
.word 0b1000000011000101
.word 0b0000001001100101
.word 0b0000000101100101
.word 0b1000000010010101
.word 0b0000000011100101
.word 0b1000000100010101
.word 0b1000001000010101
.word 0b1000000101000101
.word 0b1000000110001100
.word 0b1000001000101100
.word 0b1000000100101100
.word 0b0000001100010011
.word 0b1000000010101100
.word 0b0000001010010011
.word 0b0000000110010011
.word 0b0000000001110011
.word 0b1000000001101100
.word 0b0000001001010011
.word 0b0000000101010011
.word 0b0000001101000011
.word 0b0000000011010011
.word 0b0000001011000011
.word 0b0000000111000011
.word 0b1000001010001100
.word 0b1000001001001100
.word 0b0000001000110011
.word 0b0000000100110011
.word 0b0000001100100011
.word 0b0000000010110011
.word 0b0000001010100011
.word 0b0000000110100011
.word 0b1000000001011100
.word 0b1000000011001100
.word 0b0000001001100011
.word 0b0000000101100011
.word 0b1000000010011100
.word 0b0000000011100011
.word 0b1000000100011100
.word 0b1000001000011100
.word 0b1000000101001100
.word 0b0000000110001101
.word 0b0000001000101101
.word 0b0000000100101101
.word 0b1000001100010010
.word 0b0000000010101101
.word 0b1000001010010010
.word 0b1000000110010010
.word 0b1000000001110010
.word 0b0000000001101101
.word 0b1000001001010010
.word 0b1000000101010010
.word 0b1000001101000010
.word 0b1000000011010010
.word 0b1000001011000010
.word 0b1000000111000010
.word 0b0000001010001101
.word 0b0000001001001101
.word 0b1000001000110010
.word 0b1000000100110010
.word 0b1000001100100010
.word 0b1000000010110010
.word 0b1000001010100010
.word 0b1000000110100010
.word 0b0000000001011101
.word 0b0000000011001101
.word 0b1000001001100010
.word 0b1000000101100010
.word 0b0000000010011101
.word 0b1000000011100010
.word 0b0000000100011101
.word 0b0000001000011101
.word 0b0000000101001101
.word 0b1000000110001010
.word 0b1000001000101010
.word 0b1000000100101010
.word 0b0000001100011010
.word 0b1000000010101010
.word 0b0000001010011010
.word 0b0000000110011010
.word 0b0000000001111010
.word 0b1000000001101010
.word 0b0000001001011010
.word 0b0000000101011010
.word 0b0000001101001010
.word 0b0000000011011010
.word 0b0000001011001010
.word 0b0000000111001010
.word 0b1000001010001010
.word 0b1000001001001010
.word 0b0000001000111010
.word 0b0000000100111010
.word 0b0000001100101010
.word 0b0000000010111010
.word 0b0000001010101010
.word 0b0000000110101010
.word 0b1000000001011010
.word 0b1000000011001010
.word 0b0000001001101010
.word 0b0000000101101010
.word 0b1000000010011010
.word 0b0000000011101010
.word 0b1000000100011010
.word 0b1000001000011010
.word 0b1000000101001010
.word 0b1000000110000110
.word 0b1000001000100110
.word 0b1000000100100110
.word 0b0000001100010110
.word 0b1000000010100110
.word 0b0000001010010110
.word 0b0000000110010110
.word 0b0000000001110110
.word 0b1000000001100110
.word 0b0000001001010110
.word 0b0000000101010110
.word 0b0000001101000110
.word 0b0000000011010110
.word 0b0000001011000110
.word 0b0000000111000110
.word 0b1000001010000110
.word 0b1000001001000110
.word 0b0000001000110110
.word 0b0000000100110110
.word 0b0000001100100110
.word 0b0000000010110110
.word 0b0000001010100110
.word 0b0000000110100110
.word 0b1000000001010110
.word 0b0000000011000110
.word 0b0000001001100110
.word 0b0000000101100110
.word 0b1000000010010110
.word 0b0000000011100110
.word 0b1000000100010110
.word 0b1000001000010110
.word 0b1000000101000110
.word 0b0000000110001110
.word 0b0000001000101110
.word 0b0000000100101110
.word 0b1000001100010001
.word 0b0000000010101110
.word 0b1000001010010001
.word 0b1000000110010001
.word 0b1000000001110001
.word 0b0000000001101110
.word 0b1000001001010001
.word 0b1000000101010001
.word 0b1000001101001000
.word 0b1000000011010001
.word 0b1000001011001000
.word 0b1000000111001000
.word 0b0000001010001110
.word 0b0000001001001110
.word 0b1000001000110001
.word 0b1000000100110001
.word 0b1000001100100001
.word 0b1000000010110001
.word 0b1000001010100001
.word 0b1000000110100001
.word 0b0000000001011110
.word 0b0000000011001110
.word 0b1000001001100001
.word 0b1000000101100001
.word 0b0000000010011110
.word 0b1000000011100001
.word 0b0000000100011110
.word 0b0000001000011110
.word 0b0000000101001110

LTable:; 8b/10b RD- table
.word 0b0000001001110100
.word 0b0000000111010100
.word 0b0000001011010100
.word 0b1000001100011011
.word 0b0000001101010100
.word 0b1000001010011011
.word 0b1000000110011011
.word 0b1000001110001011
.word 0b0000001110010100
```

```
.word 0b1000001001011011
.word 0b1000000101011011
.word 0b1000001101001011
.word 0b1000000011011011
.word 0b1000001011001011
.word 0b1000000111001011
.word 0b0000000101110100
.word 0b0000000110110100
.word 0b1000001000111011
.word 0b1000000100111011
.word 0b1000001100101011
.word 0b1000000010111011
.word 0b1000001010101011
.word 0b1000000110101011
.word 0b0000001110100100
.word 0b0000001100110100
.word 0b1000001001101011
.word 0b1000000101101011
.word 0b0000001101100100
.word 0b1000000011101011
.word 0b0000001011100100
.word 0b0000000111100100
.word 0b0000001010110100
.word 0b1000001001111001
.word 0b1000000111011001
.word 0b1000001011011001
.word 0b0000001100011001
.word 0b1000001101011001
.word 0b0000001010011001
.word 0b0000000110011001
.word 0b0000001110001001
.word 0b1000001110011001
.word 0b0000001001011001
.word 0b0000000101011001
.word 0b0000001101001001
.word 0b0000000011011001
.word 0b0000001011001001
.word 0b0000000111001001
.word 0b1000000101111001
.word 0b1000000110111001
.word 0b0000001000111001
.word 0b0000000100111001
.word 0b0000001100101001
.word 0b0000000010111001
.word 0b0000001010101001
.word 0b0000000110101001
.word 0b1000001110101001
.word 0b1000001100111001
.word 0b0000001001101001
.word 0b0000000101101001
.word 0b1000001101101001
.word 0b0000000011101001
.word 0b1000001011101001
.word 0b1000000111101001
.word 0b1000001010111001
.word 0b1000001001110101
.word 0b1000000111010101
.word 0b1000001011010101
.word 0b0000001100010101
.word 0b1000001101010101
.word 0b0000001010010101
.word 0b0000000110010101
.word 0b0000001110000101
.word 0b1000001110010101
.word 0b0000001001010101
.word 0b0000000101010101
.word 0b0000001101000101
.word 0b0000000011010101
.word 0b0000001011000101
.word 0b0000000111000101
.word 0b1000000101110101
.word 0b1000000110110101
.word 0b0000001000110101
.word 0b0000000100110101
.word 0b0000001100100101
.word 0b0000000010110101
.word 0b0000001010100101
.word 0b0000000110100101
.word 0b1000001110100101
.word 0b1000001100110101
.word 0b0000001001100101
.word 0b0000000101100101
.word 0b1000001101100101
.word 0b0000000011100101
.word 0b1000001011100101
.word 0b1000000111100101
.word 0b1000001010110101
.word 0b1000001001110011
.word 0b1000000111010011
.word 0b1000001011010011
.word 0b0000001100011100
.word 0b1000001101010011
.word 0b0000001010011100
.word 0b0000000110011100
.word 0b0000001110001100
.word 0b1000001110010011
.word 0b0000001001011100
.word 0b0000000101011100
.word 0b0000001101001100
.word 0b0000000011011100
.word 0b0000001011001100
.word 0b0000000111001100
.word 0b1000000101110011
.word 0b1000000110110011
.word 0b0000001000111100
.word 0b0000000100111100
.word 0b0000001100101100
.word 0b0000000010111100
.word 0b0000001010101100
.word 0b0000000110101100
.word 0b1000001110100011
.word 0b1000001100110011
.word 0b0000001001101100
.word 0b0000000101101100
.word 0b1000001101100011
.word 0b0000000011101100
.word 0b1000001011100011
.word 0b1000000111100011
.word 0b1000001010110011
.word 0b0000001001110010
.word 0b0000000111010010
.word 0b0000001011010010
.word 0b1000001100011101
.word 0b0000001101010010
.word 0b1000001010011101
.word 0b1000000110011101
.word 0b1000001110001101
.word 0b0000001110010010
.word 0b1000001001011101
.word 0b1000000101011101
.word 0b1000001101001101
.word 0b1000000011011101
.word 0b1000001011001101
.word 0b1000000111001101
.word 0b0000000101110010
.word 0b0000000110110010
.word 0b1000001000111101
.word 0b1000000100111101
.word 0b1000001100101101
.word 0b1000000010111101
.word 0b1000001010101101
.word 0b1000000110101101
.word 0b0000001110100010
.word 0b0000001100110010
.word 0b1000001001101101
.word 0b1000000101101101
.word 0b0000001101100010
.word 0b1000000011101101
.word 0b0000001011100010
.word 0b0000000111100010
.word 0b0000001010110010
.word 0b1000001001111010
.word 0b1000000111011010
.word 0b1000001011011010
.word 0b0000001100011010
.word 0b1000001101011010
.word 0b0000001010011010
.word 0b0000000110011010
.word 0b0000001110001010
.word 0b1000001110011010
.word 0b0000001001011010
.word 0b0000000101011010
.word 0b0000001101001010
.word 0b0000000011011010
.word 0b0000001011001010
.word 0b0000000111001010
.word 0b1000000101111010
.word 0b1000000110111010
.word 0b0000001000111010
.word 0b0000000100111010
.word 0b0000001100101010
.word 0b0000000010111010
.word 0b0000001010101010
.word 0b0000000110101010
.word 0b1000001110101010
.word 0b1000001100111010
.word 0b0000001001101010
.word 0b0000000101101010
.word 0b1000001101101010
.word 0b0000000011101010
.word 0b1000001011101010
.word 0b1000000111101010
.word 0b1000001010111010
.word 0b1000001001110110
.word 0b1000000111010110
.word 0b1000001011010110
.word 0b0000001100010110
.word 0b1000001101010110
.word 0b0000001010010110
.word 0b0000000110010110
.word 0b0000001110000110
.word 0b1000001110010110
.word 0b0000001001010110
.word 0b0000000101010110
.word 0b0000001101000110
.word 0b0000000011010110
.word 0b0000001011000110
.word 0b0000000111000110
.word 0b1000000101110110
.word 0b1000000110110110
.word 0b0000001000110110
.word 0b0000000100110110
.word 0b0000001100100110
.word 0b0000000010110110
.word 0b0000001010100110
.word 0b0000000110100110
.word 0b1000001110100110
.word 0b0000001100110110
.word 0b0000001001100110
.word 0b0000000101100110
.word 0b1000001101100110
.word 0b0000000011100110
.word 0b1000001011100110
.word 0b1000000111100110
.word 0b1000001010110110
.word 0b0000001001110001
.word 0b0000000111010001
.word 0b0000001011010001
.word 0b1000001100011110
.word 0b0000001101010001
.word 0b1000001010011110
.word 0b1000000110011110
.word 0b1000001110001110
.word 0b0000001110010001
.word 0b1000001001011110
.word 0b1000000101011110
.word 0b1000001101001110
.word 0b1000000011011110
.word 0b1000001011001110
.word 0b1000000111001110
.word 0b0000000101110001
.word 0b0000000110110001
.word 0b1000001000110111
.word 0b1000000100110111
.word 0b1000001100101110
.word 0b1000000010110111
.word 0b1000001010101110
.word 0b1000000110101110
.word 0b0000001110100001
.word 0b0000001100110001
.word 0b1000001001101110
.word 0b1000000101101110
.word 0b0000001101100001
.word 0b1000000011101110
.word 0b0000001011100001
.word 0b0000000111100001
.word 0b0000001010110001

FTable:;10b/4b LUT
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
```

```
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
```

```
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
```

```
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
.word 0b0000000000000000
;756~764
.word 0b0000000000000001
.word 0b0000000000000001
.word 0b0000000000000001
.word 0b0000000000000001
.word 0b0000000000000001
.word 0b0000000000000001
.word 0b0000000000000001
.word 0b0000000000000001
.word 0b0000000000000001
; 765~773
.word 0b0000000000000010
.word 0b0000000000000010
.word 0b0000000000000010
.word 0b0000000000000010
.word 0b0000000000000010
.word 0b0000000000000010
.word 0b0000000000000010
.word 0b0000000000000010
.word 0b0000000000000010
; 774~781
.word 0b0000000000000011
.word 0b0000000000000011
.word 0b0000000000000011
.word 0b0000000000000011
.word 0b0000000000000011
.word 0b0000000000000011
.word 0b0000000000000011
.word 0b0000000000000011
;782~789
.word 0b0000000000000100
.word 0b0000000000000100
.word 0b0000000000000100
.word 0b0000000000000100
.word 0b0000000000000100
.word 0b0000000000000100
.word 0b0000000000000100
.word 0b0000000000000100
;790~794
.word 0b0000000000000101
.word 0b0000000000000101
.word 0b0000000000000101
.word 0b0000000000000101
.word 0b0000000000000101
; 795~796
.word 0b0000000000000110
.word 0b0000000000000110
; 797
.word 0b0000000000000111
; 798
.word 0b0000000000001000
; 799
.word 0b0000000000001001
; 800 ~ 801
.word 0b0000000000001010
.word 0b0000000000001010
; 802~806
.word 0b0000000000001011
.word 0b0000000000001011
.word 0b0000000000001011
.word 0b0000000000001011
.word 0b0000000000001011
; 807~814
.word 0b0000000000001100
.word 0b0000000000001100
.word 0b0000000000001100
.word 0b0000000000001100
.word 0b0000000000001100
.word 0b0000000000001100
.word 0b0000000000001100
.word 0b0000000000001100
;815~822
.word 0b0000000000001101
.word 0b0000000000001101
.word 0b0000000000001101
.word 0b0000000000001101
.word 0b0000000000001101
.word 0b0000000000001101
.word 0b0000000000001101
.word 0b0000000000001101
; 823~831
.word 0b0000000000001110
```

```
.word 0b0000000000001110
.word 0b0000000000001110
.word 0b0000000000001110
.word 0b0000000000001110
.word 0b0000000000001110
.word 0b0000000000001110
.word 0b0000000000001110
.word 0b0000000000001110
; 824～1023
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111
.word 0b0000000000001111

.end ;End of program code
```

References

[1] G. Potter, "Digital Power: A Manufacturer's View," in *Digital Power Forum proceedings*, 2004.

[2] L. Brush, "Trends in digital power management: power converter and system demand characteristics," in *Applied Power Electronics Conference and Exposition, 2005. APEC 2005. Twentieth Annual IEEE*, vol. 1, 2005, pp. 163-166.

[3] S. Buso, *Digital Control in Power Electronics*. Morgan & Claypool, 2006.

[4] H. Al-Atrash, "Digital Controller Design for a Practicing Power Electronics Engineer," in *Applied Power Electronics Conference, APEC 2007 - Twenty Second Annual IEEE*, 2007, pp. 34-41.

[5] S. Choudhury, "Designing the Digital Compensator For UCD91xx based Digital Power Supply," *Power Supply Design Seminar*, 2007.

[6] G. Feng, "A New Adaptive Fuzzy Logic Control Method for DC-to-DC Converter," in *Industrial Electronics Society, 2002, IECON '02*, 2002.

[7] A. G. Perry, "A Design Method for PI-like Fuzzy Logic Controllers for DC–DC Converter," *IEEE Transactions On Industrial Electronics*, vol. 54, no. 5, 2007.

[8] G. Feng, "A New Digital Control Algorithm to Achieve Optimal Dynamic Performance in DC-to-DC Converters," *IEEE Transactions On Power Electronics*, vol. 22, no. 4, 2007.

[9] K. Leung, "Design and implementation of a practical digital PWM controller," in *Applied Power Electronics Conference and Exposition, 2006. APEC '06. Twenty-First Annual IEEE*, 2006, p. 6.

[10] L. Balogh, "A Practical Introduction to Digital Power Supply Control," *Power Supply Design Seminar*, 2005.

[11] Y. Panov and M. M. Jovanovic, "Small-signal analysis and control design of isolated power supplies with optocoupler feedback," *Power Electronics, IEEE Transactions on*, vol. 20, no. 4, pp. 823-832, Jul. 2007.

[12] B. Chen, "Isolation in Digital Power Supplies Using Micro-Transformers," in *Applied Power Electronics Conference and Exposition, 2009. APEC 2009. Twenty-Fourth Annual IEEE*, 2009, pp. 2039-2042.

[13] B. Chen, "Isolated half-bridge gate driver with integrated high-side supply," in *Power Electronics Specialists Conference, 2008. PESC 2008. IEEE*, 2008, pp. 3615-3618.

[14] B. Chen, "Fully integrated isolated dc-dc converter using micro-transformers," in *Applied Power Electronics Conference and Exposition, 2008. APEC 2008. Twenty-Third Annual IEEE*, 2008, pp. 335-338.

[15] A. I. Pressman, *Switching Power Supply Design*. McGraw-Hill, Inc, 1991.

[16] S. Maniktala, *Switching power supply design & optimization*. McGraw-Hill Professional, 2005.

[17] D. Mitchell, "Designing Stable Control Loops," *Power Supply Design Seminar*, 2001.

[18] S. R. S. Peterchev, "Quantization Resolution and Limit Cycling in Digitally Controlled PWM Converters," *IEEE Transactions on Power Electronics*, vol. 18, pp. 301-308, 2003.

[19] "System Management Bus Specification," Revision 1.1, Intel Corporation et al., 1998.

[20] "System Management Bus Specification," Revision 2.0, Intel Corporation et al., 2000.

[21] "Digital power management protocol," Power Electronics Technology, 2004.

[22] A. Bakker, L. McGarry, and E. Miranda, "Current share using digital techniques," in *Darnell Digital Power Forum 2006 Presentation 3.2*, Richardson, Texas, 2006.

[23] B. Daly, "Digital bus makes current sharing more robust," *Power Electronics Technology*, pp. 28-33, Nov. 2006.

[24] ANSI/TIA/EIA-422-B, " Electrical Characteristics of Balanced Voltage Digital Interface Circuits," Telephone Industry Association and Electronics Industry Association, 1994.

[25] "The I2C -Bus Specification," Revision 2.1, Philips Semiconductors, Jan. 2000.

[26] "CAN Specification," Revision 2.0, Robert Bosch Gmbh, 1991.

[27] ISO 11898-1, "Road Vehicles - Controller Area Network (CAN) - Part 1: Data Link Layer And Physical Signaling," Revision 03, Dec. 2003.

[28] "LIN Specification Package," Revision 2.0, LIN Consortium, Oct. 2006.

[29] R. White, "Data Communications Issues For Power System Management," in *Applied Power Electronics Conference, APEC 2007 - Twenty Second Annual IEEE*, 2007, pp. 1188-1199.

[30] R. White, "Digital Power System Management," in *Proceedings Of The Applied Power Electronics Conference, APEC 2006*, 2006, pp. 176-182.

[31] R. White, "Considerations For A Power System Management Protocol," in *Darnell Digital Power Forum 2004, Presentation 5-1*, 2004.

[32] A. Prodic, D. Maksimovic, and R. W. Erickson, "Digital Controller Chip Set for Isolated DC Power Supplies," in *Applied Power Electronics Conference and Exposition, 2003. APEC '03. Eighteenth Annual IEEE*, vol. 2, 2003, pp. 9-13.

[33] "SMPS AC/DC Reference Design User's Guide," Microchip Technology Inc. DS70320B, 2008.

[34] "Quad-Channel Digital Isolator," Silicon Laboratories Si8440/1/2, 2006.

[35] W. Elmenreich and M. Delvai, "Time-triggered communication with UARTs," in *Proceedings of the 4th IEEE International Workshop on Factory Communication Systems (WFCS'02)*, Sweden, 2002.

[36] R. K. Orr, Y. Liu, D. A. Brown, and R. Duvnjak, "Signal and Power Transformer Coupling Arrangements," U.S. Patent US 6,911,746 B2, Jun. 28, 2005.

[37] D. J. Morris, *Pulse code formats for fiber optical data communication: basic principles and applications*, illustrated ed. CRC Press, 1983.

[38] W. Stallings, *Data and Computer Communications (7th ed.)*. Prentice Hall, 2004.

[39] D. Barrett and T. King, *Computer networking illuminated.* Jones & Bartlett Publishers, 2004.

[40] P. Oppenheimer, *Troubleshooting Campus Networks : Practical Analysis of Cisco and LAN Protocols*. New York: John Wiley & Sons, Inc., 2002.

[41] A. Widmer X, "DC-balanced 6B/8B transmission code with local parity ," Katonah, NY, US Patent United States Patent 6876315.

[42] A. Widmer, "A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code," vol. 27, no. 5, p. 440, 1983.

[43] R. C. Dorf, *Modern Control Systems (9th ed.)*. Prentice Hall, 2000.

[44] "Implementing an 8b/10b Encoder/Decoder for Gigabit Ethernet in the Actel SX FPGA Family," Actel Corporation 5192650-0/10.98, 1998.